

TTRS: Tinkoff Transactions Recommender System benchmark

Sergey Kolesnikov, Oleg Lashinin, Michail Pechatov, Alexander Kosov

Tinkoff

{s.s.kolesnikov, o.a.lashinin, m.pechatov, a.kosov}@tinkoff.ai

Russia

ABSTRACT

Over the past decade, tremendous progress has been made in inventing new RecSys methods. However, one of the fundamental problems of the RecSys research community remains the lack of applied datasets and benchmarks with well-defined evaluation rules and metrics to test these novel approaches. In this article, we present the TTRS - Tinkoff Transactions Recommender System benchmark. This financial transaction benchmark contains over 2 million interactions between almost 10,000 users and more than 1,000 merchant brands over 14 months. To the best of our knowledge, this is the first publicly available financial transactions dataset. To make it more suitable for possible applications, we provide a complete description of the data collection pipeline, its preprocessing, and the resulting dataset statistics. We also present a comprehensive comparison of the current popular RecSys methods on the next-period recommendation task and conduct a detailed analysis of their performance against various metrics and recommendation goals. Last but not least, we also introduce **Personalized Item-Frequencies-based Model (Re)Ranker** – PIFMR, a simple yet powerful approach that has proven to be the most effective for the benchmarked tasks.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

datasets, neural networks, recommender systems

ACM Reference Format:

Sergey Kolesnikov, Oleg Lashinin, Michail Pechatov, Alexander Kosov . 2021. TTRS: Tinkoff Transactions Recommender System benchmark. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

In the current era of big data and global personalization, Recommender Systems (RecSys) are playing a pivotal role in improving user experience in a large variety of domains: movies [43], music [35], news [12] and many more [5, 34, 55]. There are many different types of recommendations, such as rating prediction [23], as well as top-n [9], sequential [54], session-based [53], and next-basket

[16, 37, 40] recommendations. Thanks to close collaboration between academia and industry professionals [3, 7] on joint research topics, many new articles and methods are released every year [58]. Various new techniques aim to improve RecSys approaches with deep learning-based methods [28, 37, 56], memory-based methods [42], latent factor-based methods [9, 24, 38], or reinforcement learning [1, 29]. With such a rapid growth of new approaches, it is almost impossible to evaluate all of them correctly. Moreover, each application requires its own logic for data preprocessing and evaluation setup [32]. Last but not least, current machine learning methods require careful selection of hyperparameters, which only complicates evaluation correctness [4, 50].

To combat the challenges above, many RecSys benchmark frameworks have been created: RecBole [60], Elliot [2], ReChorus [51], DaisyRec [47]. All these benchmarks cover such popular tasks as top-n, next-item, or next-basket recommendations, which are essential for real-world applications. However, other equally valuable scenarios have received little attention in the research community. In our work, we focus on one of these scenarios. Specifically, we consider forming personalized recommendations for a certain period – the next-period recommendation task [59]. This task can have various business applications, such as monthly customers' cashback recommendations or products-of-interest list creation during users' online sessions on websites [48].

To this end, we would like to propose a large-scale financial transactions dataset and provide a comprehensive comparison of the current popular RecSys methods on the next-period recommendation task. The contributions of this paper are summarized as follows:

- Our first contribution is a large-scale financial transactions dataset – TTRS (Section 3). It contains over 2 million interactions between almost 10,000 users and more than 1,000 merchants for a total of 14 months. To the best of our knowledge, it is the first publicly available dataset of financial transactions.
- Our second contribution is benchmarking a various number of existing RecSys methods on the next-period recommendation task (Section 4). To ensure the reproducibility criteria, we carefully describe the evaluation techniques and the metrics required for correct benchmarking. In addition, we conduct a comprehensive quantitative study and comparative analysis of the different methods used in our benchmarking.
- The final point of our study is the proposal of a new approach: **Personalized Item-Frequencies-based Model (Re)Ranker**, or PIFMR. It summarises our experiments' findings and achieves the best results on the benchmarked tasks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Dataset	Before preprocessing			After preprocessing			Final statistics			
	# users	# items	# interactions	# users	# items	# interactions	# inter. per user	# inter. per user	# inter. per item in month	# inter. per item in month
TaFeng	32266	23812	817741	3470	2929	196549	56.64	14.16	67.1	16.78
Dunnhumby	50000	4997	31057875	11047	3178	11594609	1049.57	40.37	3648.4	140.32
TTRS	50000	2873	14287287	9396	1157	2744828	292.13	20.87	2372.37	169.45

Table 1: Dataset statistics before and after preprocessing

We hope that our open-sourced TTRS dataset with a described benchmark on the next-period recommendation task and the proposed PIFMR approach will serve as a foundation for other industrial research labs to develop real-world applied standards for the progress of the RecSys field.

2 RELATED WORK

Large-scale datasets and benchmarks have successively proven their fundamental importance in many research fields. RecSys was no exception, and many recent breakthroughs came with the emergence of such benchmarks [11, 30]. In this section, we observe popular datasets and methods as well as associated benchmarks and evaluation strategies.

2.1 Tasks

In general, the majority of recommender systems aim to learn users' interests. There are several specific ways we can formulate such recommendation tasks. In the classical formulation, the researchers hide some of the user's interactions for a test set. Next, the model ranks all items for each user, trying to predict the hidden interactions. This task is called the top-n recommendation task [10]. In another scenario, the model knows the user's previous ordered history of interactions and predicts the next item the user could interact with. This is the next-item prediction task [36]. If the user can consume sets of items simultaneously and we want to predict the whole set of interactions, this is called the next-basket prediction task [36]. Finally, if we are curious about user interests over time, we could predict their interactions for a predefined period, which is the next-period recommendation task [59]. In this work, we are focused on the former task. Other similar tasks, for example, click-through rate prediction [20], rating prediction [23], are not the focus of our work.

2.2 Datasets

To compare with the transactional nature of TTRS, we reviewed the available RecSys datasets for additional properties: interaction timestamp, transaction amount, and meta-information. These properties are necessary for future studies of the proposed standard. With such requirements, we found two publicly available transaction-based datasets:

- Ta-Feng¹ - a Chinese grocery store dataset that has basket-based transaction data from November 2000 to February 2001. Each transaction has a timestamp. Items with identical

timestamps are considered as one basket. This dataset is widely used for next-basket prediction research [14, 16, 25].

- Dunnhumby² - a dataset provided by Dunnhumby that contains customers transaction data over a period of 117 weeks from April 2006 to July 2008. For benchmarking purposes, we select the "Let's Get Sort-of-Real sample 50K customers" version of the dataset, which is well-known among the research community [13, 15, 16].

Statistical information on the raw datasets is summarized in Table 1.

2.3 Methods

Statistical Recommenders. Statistical Recommenders base their predictions on learned regular patterns. The simplest methods are TopPopular and TopPersonal, which we will describe in detail in the Experiments section.

General Recommenders. General Recommenders explore user feedback, which can be represented as a user-item interaction matrix. Historical data for each user is treated as an unordered collection of items, and the main goal is to predict relevance scores in missing values for future recommendations. The most known method for this task is matrix factorization [9, 17, 26, 39], which learns the hidden representations of users and items. Another possible approach is item similarity models [21, 33, 45, 52], which learn the item-item similarity matrix. Besides machine learning-based techniques, there is also a broad range of deep learning methods. For example, there are plenty of VAE-based methods [27, 28, 31, 44], which generalize linear latent-factor models with neural networks.

Sequential Recommenders. Another subfield of the RecSys methods domain is sequential recommenders. In the next-item prediction task, historical data for each user is stored as a time-ordered sequence of items users interacted with, and the goal is to predict the next relevant item. While prior methods, like First-order Markov Chains [40], assume that users' later actions depend only on the last item they interacted with, current sequential recommenders find dependencies using a complete history of user's interactions. To find such sequential patterns, CNNs [57], RNNs [49], and the self-attention mechanism [22, 46] are widely used. In the next-basket prediction task, the data is represented as a sequence of sets of purchased items, and the main goal is to predict a whole set of required items that would follow already purchased ones, rather than only one item. A wide range of basket-based methods [14, 16, 37] can be used for this task.

¹<https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset>

²<https://www.dunnhumby.com/source-files/>

column	description	# unique values	column type
party_rk	unique user identification (anonymized)	50000	int
financial_account_type_cd	account type	2	categorical int
transaction_type_desc	transaction type	4	categorical string
merchant_type	merchant type (anonymized)	464	categorical int
merchant_group_rk	merchant group identifier (anonymized)	2873	categorical int
category	merchant category	36	categorical string
transaction_dttm	transaction timestamp	-	datetime
transaction_amt	transaction amount	-	float

Table 2: TTRS dataset description. We use *party_rk* as user identifier and *merchant_group_rk* as item identifier.

2.4 Evaluation

Evaluating Recommender Systems can be challenging due to various possible data splitting strategies and data preparation approaches. In early works [6, 8], the researchers highlighted the importance of time-based algorithm validation. In a prior study [47], the authors sampled 85 papers published in 2017-2019 from top conferences and concluded that random-split-by-ratio and leave-one-out splitting strategies are used in 82% cases. At the same time, recent studies [32] pointed out that the most strict and realistic setting for data splitting is a global temporal split, where a fixed time-point separates interactions for training and testing. The authors found that only 2 of 17 recommender algorithms (published from 2009 to 2020) were evaluated using this scenario. In another work [18], the authors compare the impact of data leaks on different RecSys methods. They found that "future data" can improve or deteriorate recommendation accuracy, making the impact of data leakage unpredictable. In this paper, to avoid all the issues above, we use a global temporal K-fold validation scheme (Section 4).

3 DATASET

In this section, we describe the pipeline for collecting and processing the TTRS dataset³. We first present the raw data collection pipeline and then provide details of the data preprocessing logic and extra filtering efforts to make the data consistent for research needs.

Data Description. The crucial part of the TTRS dataset is the diversity of the transaction sources. While other datasets handle only merchant user activity, TTRS contains the whole user financial activity – supermarkets, clothing stores, online delivery shops, cinemas, gas stations, cafes and restaurants, museums, etc. Thus, TTRS contains anonymized information about the daily interests of users based on their transactional activity. To the best of our knowledge, this is the first publicly available dataset that makes it possible to build financial activity recommendations.

Data Collection. Our dataset contains transaction information of a randomly selected 50 thousand Tinkoff users from January 2019 to March 2020. Each transaction contains an anonymized user id, transaction type, information about a purchased merchant, transaction timestamp, and transaction amount. Full description of the dataset can be found in Table 2.

³The dataset is available on request to authors upon submitting a license agreement.

Data Preprocessing Pipeline. To prepare the dataset for benchmarking, we apply a few preprocessing steps. First, we truncate the dates to have full months for simplified evaluation on time periods of weeks and months. Secondly, we remove users and items with less than ten interactions in the first six months and filter users with less than one transaction per month to reduce possible anomalies. As the number of interactions between the remaining users and items could change after filtering, we repeat the second step several times until the data converges. Statistic information about clean datasets after preprocessing is summarized in Table 1.

4 EXPERIMENTS

For our benchmark, we chose the next-period recommendation task with a period of one month. The main goal of our benchmark was to predict users' interactions in the next month, using their interaction history over the past few months. In this section, we will go through the experiment setup, benchmarked methods and introduce our main findings and improvements.

4.1 Evaluation Setup

Metrics. We compare models with each other using standard metrics widely utilized by researchers: Recall@K, NDCG@K, and MAP@K. Each metric can be calculated for a recommendation list of length K, where K ranges from 1 to the number of items. K is usually called the cutoff, which stands for the length to which the recommendations are cut. We use @10, @20, @50 cutoffs during benchmarking.

Validation Scheme. To get the most accurate results, we use several ideas from prior articles [6, 11, 32]. Firstly, we use a global temporal split to separate our training data from test one and prevent possible data leaks associated with seasonal user preferences. Secondly, we use temporal cross-validation with several folds for a more precise model evaluation. Finally, for each such fold with N periods, we use the optimal hyperparameter search through extra data partitioning into "train" (N-2 periods), "validation" (1 period), and "test" (1 period) splits. The best hyperparameters found on the ("train", "validation") split were used to initialize and train the model for final testing on the ("train+validation", "test") split. The entire validation process is shown in Figure 1

Hyperparameter Search. Similar to previous studies [11], we search for the optimal parameters through Bayesian search using the implementation of Scikit-Optimize⁴. For each pair (algorithm, test

⁴<https://scikit-optimize.github.io/>

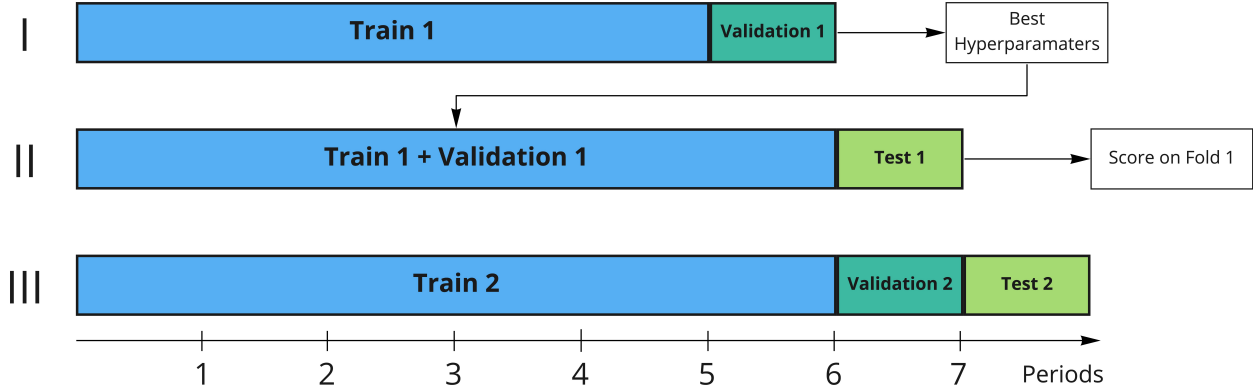


Figure 1: Evaluation procedure. On step (I), we train model on the (“Train1”, “Validation1”) split to find optimal hyperparameters for the first fold. On step (II), we use these hyperparameters to train the model on the (“Train 1 + Validation 1”, “Test 1”) split to evaluate the model on the first fold. On step (III), we repeat the (I) and (II) steps for the next fold.

fold), we iterate over 25 hypotheses, where the first 5 are random. We use $MAP@10$ metric for model selection. We share all the code as well as details of the respective hyperparameter ranges and the final algorithms’ hyperparameters online ⁵.

Computational Resources. The experiments were run on a single machine with NVIDIA Tesla V100, 200GB RAM, and Intel(R) Xeon(R) CPU @ 3.00GHz (16 cores) for 5 days.

4.2 Methods

In this section, we would like to briefly describe approaches used for benchmarking next-period recommendations:

- **TopPopular** [19], **TopPersonal** [16] are simple RecSys baselines that work based on item popularity. The TopPopular algorithm recommends the most popular items, sorting them in descending order of global popularity. The TopPersonal method focuses on items with which the user has already interacted. The recommendation list is created by sorting them in descending order of interaction frequency. If no personal recommendations are found, TopPersonal uses the TopPopular approach for prediction.
- **NMF** [26], **PureSVD** [9], **IALS** [17] are matrix factorization-based (MF-based) models. These models are designed to approximate any value in the interaction matrix by multiplying the user and item vectors in the hidden space. The interaction matrix could be represented as a matrix with interaction frequencies or in binary form. We use *binary_matrix* hyperparameter to handle such data preprocessing. In the first case, we apply the $\log(1+p)$ transformation. In another case, all frequencies above 0 were replaced by 1.
- **SLIM** [33], **EASE** [45] are linear models that learn an item-item weight matrix. Similar to MF-based models, we add a hyperparameter *binary_matrix* to approximate frequencies or a binary mask of interactions.

- **Multi-VAE** [28], **Macrid-VAE** [31], **RecVAE** [44] are variational autoencoder approaches (VAE-based) for a top-n recommendation task. They utilize the idea of using multinomial likelihood to recover inputs from hidden representations and use them for recommendations.
- **GRU4Rec** [49], **SASRec** [22], **BERT4Rec** [46] are sequential-based models. Unlike previously mentioned methods, these models know the sequence of users’ interactions and learn sequence representation with RNN- or self-attention-based neural networks. This representation is then used for next-item recommendations.
- **RepeatNet** [37] is an RNN-based model that uses a repeat-explore mechanism for session-based recommendations. The model has two different recommendation modes. In the first, “repeat” mode, the model recommends something from users’ consumption history. In the second, “explore” mode, the model recommends something new that hasn’t been listed in the input sequence.
- **ItemKNN** [41] is an item-based k-nearest neighbors method, which utilizes similarities between previously purchased items. Similar to [11], we used different similarity measures during our experiments: Jaccard coefficient, Cosine, Asymmetric Cosine, and Tversky similarity.
- **TIFUKNN** [16] is a current state-of-the-art method for the next-basket recommendation task. It uses the idea of learning Temporal Item Frequencies with the k-nearest neighbors approach.

To summarize, we consider 16 models, 7 of which are neural networks, and 5 are sequence-aware. For benchmarking purposes, we include approaches of different types such as matrix factorization, linear models, variational autoencoders, recurrent neural networks, and self-attention-based methods.

We adapted all the above models for the next-period recommendation task. To predict the next period (one month in our case), we use all available history for *Statistical* or *General Recommenders*

⁵We intend to release the code after the paper is accepted

Dataset	Model	Type	RECALL			NDCG			MAP			Benchmark time
			K@10	K@20	K@50	K@10	K@20	K@50	K@10	K@20	K@50	
Dunnhumby	TopPopular	Statistical	9.38	13.17	19.88	15.28	18.16	22.22	16.48	10.85	8.33	3 min
	TopPersonal		22.12	31.36	44.37	32.38	39.45	47.29	44.73	33.38	27.5	6 min
	NMF	MF	19.14	26.06	34.56	29.34	34.65	39.8	39.7	27.71	21.3	3 h 46 min
	PureSVD		19.43	27.19	38.0	29.56	35.45	42.03	40.0	28.43	22.75	8 min
	IALS		15.17	21.45	29.03	21.56	26.38	30.95	26.14	18.64	14.13	2 h 46 min
	SLIM	I2I	12.57	18.79	30.18	19.02	23.64	30.39	21.13	15.05	12.67	1 h 42 min
	EASE		11.95	18.34	29.89	17.56	22.28	29.16	19.07	13.98	12.12	9 min
	MultiVAE	VAE	13.15	20.1	31.96	19.06	24.23	31.29	21.65	15.88	13.47	1 h 59 min
	MacridVAE		21.24	30.15	41.72	30.92	37.75	44.89	42.03	31.25	25.36	2 h 33 min
	RecVAE		21.43	29.69	40.76	31.17	37.54	44.35	42.63	31.11	24.95	15 h 37 min
	GRU4Rec	Sequential	10.96	15.09	22.45	17.34	20.42	24.78	17.97	12.1	9.92	6 h 45 min
	SASRec		15.13	20.42	29.0	22.95	26.89	32.03	25.95	17.9	14.84	23 h 28 min
	BERT4Rec		15.37	20.79	29.67	23.55	27.62	32.95	26.78	18.5	15.27	22 h 3 min
	RepeatNet	Sequential	22.27	31.53	44.73	32.57	39.66	47.64	45.08	33.67	27.78	10 h 13 min
	ItemKNN	KNN	10.32	15.36	24.92	16.26	20.1	25.9	17.89	12.4	10.14	24 min
	TIFUKNN		20.98	29.61	44.16	30.87	37.41	46.3	41.27	30.38	26.12	8 h 11 min
	SLIM	PIFMR	22.68	31.62	38.85	33.05	39.97	44.49	46.19	34.2	25.97	57 min
	EASE		22.67	32.58	46.06	33.03	40.59	48.84	46.14	34.84	28.91	9 min
	MultiVAE		22.66	32.54	46.05	33.03	40.56	48.83	46.13	34.8	28.9	1 h 40 min
	ItemKNN		16.5	24.85	36.36	24.24	30.64	38.05	30.01	22.77	19.36	30 min
TTRS	TopPopular	Statistical	28.79	45.17	59.07	31.18	38.58	44.32	20.86	21.64	23.79	1 min
	TopPersonal		60.5	71.96	82.92	63.41	69.65	74.52	54.98	54.18	56.82	7 min
	NMF	MF	59.34	70.17	75.55	62.53	68.5	70.93	53.95	52.98	54.18	5 h 40 min
	PureSVD		59.6	70.67	75.81	62.72	68.8	71.17	54.15	53.27	54.48	3 min
	IALS		49.91	66.94	77.04	45.22	54.2	58.62	33.34	35.91	38.19	2 h 37 min
	SLIM	I2I	39.16	54.37	72.99	39.4	46.89	54.41	28.13	28.93	32.23	18 min
	EASE		32.48	47.28	68.77	33.65	40.91	49.18	22.98	23.76	27.07	3 min
	MultiVAE	VAE	50.06	67.28	<u>83.47</u>	47.13	56.04	62.94	35.58	37.71	41.34	1 h 44 min
	MacridVAE		52.64	62.87	76.15	57.09	62.54	68.16	47.22	46.05	48.68	1 h 23 min
	RecVAE		54.74	65.39	78.19	58.95	64.64	70.08	49.48	48.35	51.0	3 h 45 min
	GRU4Rec	Sequential	58.65	68.97	80.14	62.66	68.29	73.1	53.75	52.68	55.08	12 h 6 min
	SASRec		54.01	64.48	76.94	58.12	63.74	69.06	48.47	47.59	50.12	9 h 8 min
	BERT4Rec		55.86	66.68	79.06	59.03	64.84	70.15	49.48	48.76	51.37	20 h 33 min
	RepeatNet	Sequential	61.15	<u>72.74</u>	82.49	<u>63.9</u>	<u>70.26</u>	<u>74.6</u>	<u>55.53</u>	<u>54.85</u>	<u>57.21</u>	7 h 33 min
	ItemKNN	KNN	31.81	43.32	57.78	33.9	39.93	46.1	23.69	23.85	26.47	13 min
	TIFUKNN		59.96	71.93	81.3	62.98	69.6	73.79	54.09	53.85	56.13	4 h 35 min
	SLIM	PIFMR	61.39	73.53	84.19	64.01	70.67	75.36	55.67	55.24	57.8	8 min
	EASE		61.31	73.43	83.94	63.96	70.6	75.24	55.61	55.17	57.71	3 min
	MultiVAE		61.34	73.42	84.2	63.97	70.61	75.35	55.62	55.18	57.77	1 h 12 min
	ItemKNN		53.82	64.97	75.8	55.77	61.91	66.58	45.79	45.6	47.9	15 min
TaFeng	TopPopular	Statistical	4.77	8.14	14.15	5.58	7.32	9.84	3.29	2.92	3.11	1 min
	TopPersonal		13.11	17.13	23.89	15.38	17.59	20.46	<u>10.62</u>	9.14	9.21	4 min
	NMF	MF	11.29	15.35	20.1	12.91	15.19	17.37	8.42	7.35	7.42	1 h 8 min
	PureSVD		12.42	17.91	24.1	13.8	16.82	19.66	9.05	8.14	8.38	3 min
	IALS		11.55	18.15	25.86	11.14	14.66	18.16	6.4	6.26	6.84	2 h 33 min
	SLIM	I2I	11.91	17.35	25.62	12.84	15.75	19.34	8.55	7.55	7.89	10 min
	EASE		13.58	19.42	27.31	14.4	17.5	20.98	9.57	8.55	8.88	3 min
	MultiVAE	VAE	9.91	15.87	24.38	9.45	12.53	16.27	5.29	5.17	5.75	1 h 7 min
	MacridVAE		13.04	18.81	27.68	14.48	17.52	21.36	9.65	8.66	9.1	1 h 13 min
	RecVAE		13.0	18.5	25.98	14.76	17.68	21.0	9.96	8.86	9.18	2 h 9 min
	GRU4Rec	Sequential	5.06	8.78	14.95	5.14	7.0	9.63	2.82	2.6	2.88	1 h 56 min
	SASRec		10.06	14.5	22.4	10.78	13.08	16.39	6.57	6.02	6.41	3 h 30 min
	BERT4Rec		6.84	10.07	16.55	7.87	9.57	12.24	4.82	4.35	4.59	4 h 17 min
	RepeatNet	Sequential	<u>13.76</u>	19.29	27.36	<u>15.58</u>	<u>18.57</u>	<u>22.05</u>	10.59	<u>9.39</u>	<u>9.7</u>	1 h 49 min
	ItemKNN	KNN	7.25	10.93	14.87	7.31	9.35	11.18	4.29	3.91	4.03	10 min
	TIFUKNN		13.44	<u>19.54</u>	<u>27.73</u>	15.09	18.33	21.99	10.01	9.11	9.56	4 h 10 min
	SLIM	PIFMR	14.38	20.23	28.76	16.41	19.53	23.23	11.33	10.09	10.41	9 min
	EASE		14.22	19.86	27.8	16.2	19.22	22.72	11.21	9.93	10.23	3 min
	MultiVAE		14.33	20.01	28.58	16.29	19.34	23.04	11.18	9.91	10.23	35 min
	ItemKNN		10.74	15.17	20.85	10.73	13.18	15.71	6.45	5.91	6.11	12 min

Table 3: Next-month recommendation benchmark. Ground truth test interactions are not necessarily new to users, and recommendation lists are not filtered in any way. All metrics are averaged over 2 (TaFeng) or 6 (TTRS and Dunnhumby) test folds. Benchmark time is estimated as the overall time required for a hyperparameter search (25 hypotheses) and all included metrics calculations.

Algorithm 1 Personalized Item Frequency Model (Re)Ranker

```

1:  $MinFreq$  – hyperparameter
2:  $F$  – personalized frequency-based item statistics
3:  $S$  – user-item scores from the model
4:  $c = 1$  - normalization constant
5:  $F_{freq} = F > MinFreq$ 
6:  $S_{max} = \max(S), S_{min} = \min(S)$ 
7:  $S_{01} = (S - S_{min} + c) / (S_{max} - S_{min} + 2 * c)$ 
8:  $S_{PIFMR} = S_{01} + F_{freq}$ 

```

and 100 last interactions for *Sequential Recommenders*. All models produced predictions in the form of sorted lists of items, and were evaluated on the next-month recommendation task in the same manner.

4.3 Results

The results of the model's comparison can be found in Table 3. Several observations can be made based on these results. First, the simple baselines based on personalized items' popularity are compatible with other methods across all datasets. This indicates the importance of the user's repeat consumption pattern for recommendations.

Second, many current RecSys methods of different types (MF, linear, VAEs, sequential) could hardly beat the TopPersonal baseline across all metrics and datasets. We believe the reason for that is that for these approaches, the repetitive nature of the datasets is too difficult to generalize.

Third, recently proposed deep learning methods, such as RepeatNet, achieve performance comparable with the TopPersonal baseline. We believe the reason is that the "repeat" mode of the proposal network helps generalize the datasets' repetitive nature.

4.4 Personalized Item-Frequencies-based Model (Re)Ranker

Analyzing benchmark results, it is easy to notice the importance of repetitive user interactions with items. For example, the TopPersonal baseline often gives better results than recent machine learning approaches, where only one method out of 14 could achieve a better MAP@10 score. However, note that TopPersonal also has several disadvantages: (1) it cannot correctly rank items with the same consumption frequency, (2) it is unable to utilize users' interaction history to identify novel items for recommendation.

To overcome these issues, we propose a simple yet powerful improvement by introducing **Personalized Item-Frequencies-based Model (Re)Ranker** - PIFMR. Suppose we have a history of user interaction with items, and we can aggregate these interactions into a vector $F = (f_{u1}, \dots, f_{um})$ where f_{ui} is the number of times that item i was purchased by a user u and m is the number of items. Recommending new items by these frequencies will bring us to a TopPersonal-like approach. On the other hand, using the same interaction history, we could train a simple model such as EASE, SLIM, MF, or VAE and form their prediction vector as $S = (s_{u1}, \dots, s_{um})$ where s_{ui} is a relevance score for a user-item pair (u, i) . The main idea of PIFMR is to use Personalized Item Frequencies (vector F) to re-rank model predictions (vector S). We also perform

a monotonic transformation on the model's scores so that they lie in the $(0, 1)$ interval. As a result of this, for any pairs of items k and r PIFMR gives such predictions p , where $p_{uk} < p_{ur}$ if $f_{uk} < f_{ur}$. The final algorithm is presented in Algorithm 1. Retrained PIFMR-based models could be found in Table 3, "PIFMR" type.

Combining predictions with a PIFMR-based model, we solved several challenges at once: (1) the model is capable of learning how to rank items with the same consumption frequency, (2) thanks to base model usage, we could find patterns in the users' behavior and recommend new personalized items, (3) the model easily identifies high-frequency repetitive patterns in consumption history, (4) as well as robust low-frequency purchase "anomalies", thanks to the $MinFreq$ threshold.

4.5 New Pattern Finding Analysis

When it comes to the importance of repeated patterns, it is also interesting to investigate the benchmarked methods' ability to find new relevant items for users that are not yet present in their history. To do so, we test the same *trained* models on a slightly different task. Rather than analyze the method's performance across all possible items, both new and repeated for that user, we measure its efficiency on new items only. The results of the models' comparison for this recommendation goal can be found in Table 4.

A few interesting observations can be made based on our results. First, while TIFUKNN is showing average results across datasets, it does the best on the TaFeng one. A possible reason for this could be that the amount of data in the TaFeng dataset is rather small compared to the other ones. Second, TTRS is the only dataset where the TopPopular approach still achieves competitive results that could show the similarity of user interests in this dataset. Third, MF-based methods show the worst performance on this task across all datasets. Fourth, RepeatNet performs worse than TopPopular, which may indicate that it overfits to the "repeat" mode rather than "explore". Lastly, while the proposed PIFMR approach usually slightly worsens the performance of the MultiVAE, EASE, and SLIM methods on this task, it actually improves the performance of the ItemKNN approach (and SLIM on the TaFeng dataset). In addition to the Table 3 results, the above findings may serve as a good reason for further research on the methodology of PIF-based recommendations.

5 CONCLUSION

In this paper, we proposed a large-scale financial transactions benchmark named TTRS that is based on user-merchant interactions. We evaluated various RecSys methods on several transaction-based datasets to compare the effects of different factors on the next-period recommendation task. As shown by the benchmark, the user consumption repeatability factor is ubiquitous in many real-world applications and challenging for current RecSys methods. With this new benchmark, we also presented a simple yet powerful approach: **Personalized Item-Frequencies-based Model (Re)Ranker**, or PIFMR, which helped in improving the performance of RecSys methods on benchmarked tasks.

REFERENCES

- [1] M. Mehdi Afsar, Trafford Crump, and Behrouz H. Far. 2021. Reinforcement learning based recommender systems: A survey. *CoRR* abs/2101.06286 (2021).

Dataset	Model	Type	RECALL			NDCG			MAP		
			K@10	K@20	K@50	K@10	K@20	K@50	K@10	K@20	K@50
Dunnhumby	TopPopular	Statistical	4.63	7.09	12.13	5.83	7.34	9.78	3.74	2.82	2.76
	TopPersonal		4.63	7.09	12.13	5.83	7.34	9.78	3.74	2.82	2.76
	NMF	MF	2.41	3.93	7.13	2.81	3.7	5.21	1.55	1.24	1.26
	PureSVD		2.15	3.4	6.39	2.6	3.34	4.73	1.46	1.15	1.16
	IALS		2.66	4.67	9.35	3.06	4.27	6.5	1.69	1.38	1.46
	SLIM	I2I	5.69	9.03	15.73	6.85	8.84	12.01	4.32	3.42	3.46
	EASE		4.2	6.62	11.67	5.12	6.59	9.02	3.19	2.46	2.44
	MultiVAE	VAE	4.77	7.65	13.71	5.81	7.53	10.4	3.58	2.81	2.85
	MacridVAE		4.24	6.54	11.34	5.32	6.73	9.03	3.37	2.55	2.49
	RecVAE		4.42	6.82	11.98	5.5	6.96	9.43	3.41	2.63	2.61
	GRU4Rec	Sequential	3.71	6.05	10.92	4.38	5.75	8.03	2.52	2.01	2.07
	SASRec		3.72	6.09	11.38	4.4	5.8	8.27	2.6	2.07	2.13
	BERT4Rec		3.93	6.51	12.24	4.61	6.14	8.82	2.69	2.17	2.26
	RepeatNet	Sequential	1.88	3.02	6.12	2.42	3.11	4.58	1.44	1.05	1.04
	ItemKNN	KNN	<u>6.18</u>	<u>9.16</u>	14.67	<u>8.49</u>	<u>10.47</u>	<u>13.39</u>	<u>6.09</u>	<u>4.38</u>	<u>4.07</u>
	TIFUKNN		3.24	5.0	8.63	4.09	5.18	6.95	2.65	2.0	1.94
TTRS	SLIM	PIFMR	5.07	7.89	13.88	6.2	7.91	10.76	3.93	3.03	3.02
	EASE		5.44	8.58	15.1	6.67	8.56	11.66	4.26	3.32	3.34
	MultiVAE		5.19	8.15	14.41	6.39	8.19	11.15	4.06	3.16	3.17
	ItemKNN		6.25	9.3	15.17	8.56	10.6	13.69	6.23	4.48	4.16
	TopPopular	Statistical	17.98	27.42	42.97	13.08	16.41	20.8	7.52	8.44	9.35
	TopPersonal		17.98	27.42	42.97	13.07	16.41	20.8	7.52	8.44	9.35
	NMF	MF	3.31	4.84	7.87	2.45	2.99	3.86	1.38	1.49	1.6
	PureSVD		2.84	4.1	6.95	2.14	2.59	3.4	1.21	1.3	1.4
	IALS		9.83	15.55	26.88	6.85	8.83	11.96	3.77	4.25	4.77
	SLIM	I2I	21.98	32.52	50.19	16.1	19.79	24.72	9.62	10.71	11.78
	EASE		21.09	31.92	51.47	15.41	19.23	24.66	9.04	10.16	11.36
	MultiVAE	VAE	22.47	33.57	<u>52.75</u>	16.38	20.28	25.61	9.77	10.92	12.1
	MacridVAE		19.13	29.04	46.83	14.02	17.52	22.46	8.22	9.2	10.24
	RecVAE		19.59	29.69	47.77	14.31	17.86	22.9	8.4	9.41	10.47
	GRU4Rec	Sequential	18.3	28.4	47.72	12.66	16.19	21.53	7.26	8.24	9.34
	SASRec		20.12	30.48	48.59	14.47	18.11	23.15	8.49	9.54	10.61
	BERT4Rec		19.79	30.11	48.45	14.03	17.66	22.75	8.22	9.26	10.32
	RepeatNet	Sequential	17.24	26.17	40.46	12.23	15.37	19.36	6.93	7.79	8.57
TaFeng	ItemKNN	KNN	<u>25.33</u>	<u>34.65</u>	47.14	<u>22.18</u>	<u>26.01</u>	<u>30.14</u>	<u>13.46</u>	<u>14.59</u>	<u>15.68</u>
	TIFUKNN		14.64	22.27	35.85	10.67	13.36	17.13	6.17	6.92	7.67
	SLIM	PIFMR	21.63	32.27	50.95	15.88	19.61	24.82	9.48	10.57	11.71
	EASE		21.62	31.88	48.08	15.86	19.47	23.99	9.47	10.53	11.5
	MultiVAE		21.37	32.26	51.13	15.58	19.41	24.68	9.22	10.35	11.51
	ItemKNN		26.07	37.15	54.01	22.45	26.91	32.35	13.51	14.84	16.31
	TopPopular	Statistical	3.55	6.24	12.0	4.08	5.38	7.6	2.27	2.21	2.46
	TopPersonal		3.55	6.24	11.99	4.08	5.37	7.59	2.27	2.21	2.46
	NMF	MF	1.81	2.65	4.44	1.81	2.23	2.92	0.88	0.81	0.86
	PureSVD		1.72	2.71	5.21	1.71	2.21	3.13	0.82	0.79	0.87
	IALS		2.51	3.95	7.12	2.27	2.96	4.16	1.05	1.03	1.15
	SLIM	I2I	<u>4.35</u>	6.5	10.9	4.25	5.29	6.98	2.15	2.04	2.2
	EASE		3.76	5.9	10.02	3.55	4.59	6.17	1.75	1.71	1.86
	MultiVAE	VAE	3.29	5.12	9.36	3.04	3.92	5.54	1.46	1.44	1.6
	MacridVAE		4.25	<u>6.53</u>	11.7	4.33	5.39	7.38	2.27	2.18	2.38
	RecVAE		2.56	<u>4.21</u>	7.94	2.35	3.14	4.58	1.12	1.1	1.24
	GRU4Rec	Sequential	3.16	6.35	11.26	3.02	4.48	6.39	1.47	1.54	1.76
	SASRec		3.7	6.24	11.6	3.23	4.41	6.41	1.51	1.57	1.81
	BERT4Rec		2.44	4.4	9.14	2.14	3.05	4.82	0.98	1.0	1.19
	RepeatNet	Sequential	3.36	5.46	9.84	2.67	3.65	5.34	1.17	1.17	1.33
TaFeng	ItemKNN	KNN	2.33	3.51	5.98	2.43	3.05	4.05	1.25	1.11	1.17
	TIFUKNN		4.18	6.11	11.27	<u>4.79</u>	<u>5.72</u>	<u>7.69</u>	<u>2.67</u>	<u>2.49</u>	<u>2.69</u>
	SLIM	PIFMR	4.43	7.27	13.09	4.94	6.29	8.53	2.75	2.65	2.9
	EASE		4.09	6.51	11.0	4.09	5.25	6.96	2.1	2.04	2.21
	MultiVAE		3.78	6.84	12.55	4.09	5.55	7.79	2.21	2.19	2.44
	ItemKNN		4.87	6.88	11.71	4.61	5.66	7.63	2.33	2.14	2.32

Table 4: Next-month new item recommendation benchmark. Ground truth objects are new to users, and recommendation lists do not contain items interacted with during training. All metrics are averaged over 2 (TaFeng) or 6 (TTRS and Dunnhumby) test folds.

- arXiv:2101.06286 <https://arxiv.org/abs/2101.06286>
- [2] Vito Walter Anelli, Alejandro Bellogin, Antonio Ferrara, Daniele Malatesta, Felice Antonio Merra, Claudio Pomo, Francesco M. Donini, and Tommaso Di Noia. 2021. Elliot: a Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. *CoRR* abs/2103.02590 (2021).
 - [3] Vito Walter Anelli, Amra Delic, Gabriele Sottocornola, Jessie Smith, Nazareno Andrade, Luca Belli, Michael M. Bronstein, Akshay Gupta, Sofia Ira Ktena, Alexandre Lung-Yut-Fong, Frank Portman, Alykhan Tejani, Yuanpu Xie, Xiao Zhu, and Wenzhe Shi. 2020. RecSys 2020 Challenge Workshop: Engagement Prediction on Twitter's Home Timeline. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 623–627. <https://doi.org/10.1145/3383313.3411532>
 - [4] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Claudio Pomo, and Azzurra Ragone. 2019. On the discriminative power of Hyper-parameters in Cross-Validation and how to choose them. *CoRR* abs/1909.02523 (2019). arXiv:1909.02523 <https://arxiv.org/abs/1909.02523>
 - [5] Khalid Anwar and Shahab Sohail. 2019. Machine Learning Techniques for Book Recommendation: An Overview. *SSRN Electronic Journal* (01 2019). <https://doi.org/10.2139/ssrn.3356349>
 - [6] Joeran Beel. 2017. It's Time to Consider "Time" when Evaluating Recommender-System Algorithms [Proposal]. (08 2017).
 - [7] James Bennett and Stan Lanning. 2007. The netflix prize. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*. ACM.
 - [8] Pedro G. Campos, Fernando Diez, and Manuel Sánchez-Montañés. 2011. Towards a More Realistic Evaluation: Testing the Ability to Predict Future Tastes of Matrix Factorization-Based Recommenders. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (Chicago, Illinois, USA) (*RecSys '11*). Association for Computing Machinery, New York, NY, USA, 309–312. <https://doi.org/10.1145/2043932.2043990>
 - [9] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (Barcelona, Spain) (*RecSys '10*). Association for Computing Machinery, New York, NY, USA, 39–46. <https://doi.org/10.1145/1864708.1864721>
 - [10] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (Barcelona, Spain) (*RecSys '10*). Association for Computing Machinery, New York, NY, USA, 39–46. <https://doi.org/10.1145/1864708.1864721>
 - [11] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 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, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 101–109. <https://doi.org/10.1145/3298689.3347058>
 - [12] Chong Feng, Muzammil Khan, Arif Ur Rahman, and Arshad Ahmad. 2020. News Recommendation Systems - Accomplishments, Challenges Future Directions. *IEEE Access* 8 (2020), 16702–16725. <https://doi.org/10.1109/ACCESS.2020.2967792>
 - [13] Valeria Fionda and Giuseppe Pirrò. 2019. Triple2Vec: Learning Triple Embeddings from Knowledge Graphs. *CoRR* abs/1905.11691 (2019). arXiv:1905.11691 <http://arxiv.org/abs/1905.11691>
 - [14] Haoji Hu and Xiangnan He. 2019. Sets2Sets: Learning from Sequential Sets with Neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) (*KDD '19*). Association for Computing Machinery, New York, NY, USA, 1491–1499. <https://doi.org/10.1145/3292500.3330979>
 - [15] Haoji Hu and Xiangnan He. 2019. Sets2Sets: Learning from Sequential Sets with Neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) (*KDD '19*). Association for Computing Machinery, New York, NY, USA, 1491–1499. <https://doi.org/10.1145/3292500.3330979>
 - [16] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling Personalized Item Frequency Information for Next-basket Recommendation. 1071–1080. <https://doi.org/10.1145/3397271.3401066>
 - [17] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*, 263–272. <https://doi.org/10.1109/ICDM.2008.22>
 - [18] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. On Offline Evaluation of Recommender Systems. *CoRR* abs/2010.11060 (2020). arXiv:2010.11060 <https://arxiv.org/abs/2010.11060>
 - [19] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A Re-Visit of the Popularity Baseline in Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) (*SIGIR '20*). Association for Computing Machinery, New York, NY, USA, 1749–1752. <https://doi.org/10.1145/3397271.3401233>
 - [20] ZHOU Ao-ying, JI Wen-di, WANG Xiao-ling. 2013. Techniques for estimating click-through rates of Web advertisements: A survey. <http://hdsfdxzk.xml-journal.net/en/article/id/24855>
 - [21] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored Item Similarity Models for Top-N Recommender Systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, Illinois, USA) (*KDD '13*). Association for Computing Machinery, New York, NY, USA, 659–667. <https://doi.org/10.1145/2487575.2487589>
 - [22] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. *2018 IEEE International Conference on Data Mining (ICDM)* (2018), 197–206.
 - [23] Zahid Khan, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince. 2021. Deep learning techniques for rating prediction: a survey of the state-of-the-art. *Artificial Intelligence Review* 54 (01 2021). <https://doi.org/10.1007/s10462-020-09892-9>
 - [24] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 77–118. https://doi.org/10.1007/978-1-4899-7637-6_3
 - [25] Duc-Trong Le, Hady W. Lauw, and Yuan Fang. 2019. Correlation-Sensitive Next-Basket Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 2808–2814. <https://doi.org/10.24963/ijcai.2019/389>
 - [26] Tao Li, Jiandong Wang, Huiping Chen, Xinyu Feng, and Feiyue Ye. 2006. A NMF-based collaborative filtering recommendation algorithm. In *2006 6th World Congress on Intelligent Control and Automation*, Vol. 2. IEEE, 6082–6086.
 - [27] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) (*KDD '17*). Association for Computing Machinery, New York, NY, USA, 305–314. <https://doi.org/10.1145/3097983.3098077>
 - [28] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) (*WWW '18*). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 689–698. <https://doi.org/10.1145/3178876.3186150>
 - [29] Sam Lobel, Chunyuan Li, Jianfeng Gao, and Lawrence Carin. 2019. Towards Amortized Ranking-Critical Training for Collaborative Filtering. *CoRR* abs/1906.04281 (2019). arXiv:1906.04281 <http://arxiv.org/abs/1906.04281>
 - [30] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Empirical Analysis of Session-Based Recommendation Algorithms. *CoRR* abs/1910.12781 (2019). arXiv:1910.12781 <http://arxiv.org/abs/1910.12781>
 - [31] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2019/file/a2186aa7c086b46ad4e8bf81e2a3a19b-Paper.pdf>
 - [32] Zhaqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2020. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) (*RecSys '20*). Association for Computing Machinery, New York, NY, USA, 681–686. <https://doi.org/10.1145/3383313.3418479>
 - [33] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *2011 IEEE 11th International Conference on Data Mining*, 497–506. <https://doi.org/10.1109/ICDM.2011.134>
 - [34] V Sree Parvathy and T K Ratheesh. 2017. Friend recommendation system for online social networks: A survey. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Vol. 2. 359–365. <https://doi.org/10.1109/ICECA.2017.8212834>
 - [35] Dip Paul and Subhradeep Kundu. 2020. A Survey of Music Recommendation Systems with a Proposed Music Recommendation System. In *Emerging Technology in Modelling and Graphics*, Jyotsna Kumar Mandal and Debika Bhattacharya (Eds.). Springer Singapore, Singapore, 279–285.
 - [36] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4, Article 66 (July 2018), 36 pages. <https://doi.org/10.1145/3190616>
 - [37] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (07 2019), 4806–4813. <https://doi.org/10.1609/aaai.v33i01.33014806>
 - [38] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu (Eds.). IEEE Computer Society, 995–1000. <https://doi.org/10.1109/ICDM.2010.127>
 - [39] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal,

- Quebec, Canada) (*UAI '09*). AUAI Press, Arlington, Virginia, USA, 452–461.
- [40] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-Basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) (*WWW '10*). Association for Computing Machinery, New York, NY, USA, 811–820. <https://doi.org/10.1145/1772690.1772773>
- [41] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (Hong Kong, Hong Kong) (*WWW '01*). Association for Computing Machinery, New York, NY, USA, 285–295. <https://doi.org/10.1145/371920.372071>
- [42] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko (Eds.). ACM, 285–295. <https://doi.org/10.1145/371920.372071>
- [43] Nisha Sharma and Mala Dutta. 2020. Movie Recommendation Systems: A Brief Overview. In *Proceedings of the 8th International Conference on Computer and Communications Management* (Singapore, Singapore) (*ICCCM '20*). Association for Computing Machinery, New York, NY, USA, 59–62. <https://doi.org/10.1145/3411174.3411194>
- [44] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) (*WSDM '20*). Association for Computing Machinery, New York, NY, USA, 528–536. <https://doi.org/10.1145/3336191.3371831>
- [45] Harald Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference* (San Francisco, CA, USA) (*WWW '19*). Association for Computing Machinery, New York, NY, USA, 3251–3257. <https://doi.org/10.1145/3308558.3313710>
- [46] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) (*CIKM '19*). Association for Computing Machinery, New York, NY, USA, 1441–1450. <https://doi.org/10.1145/3357384.3357895>
- [47] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In *Proceedings of the 14th ACM Conference on Recommender Systems*.
- [48] Jacopo Tagliabue, Ciro Greco, Jean-François Roy, Binqing Yu, Patrick Chia, Federico Bianchi, and Giovanni Cassani. 2021. SIGIR 2021 E-Commerce Workshop Data Challenge.
- [49] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-Based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) (*DLRS 2016*). Association for Computing Machinery, New York, NY, USA, 17–22. <https://doi.org/10.1145/2988450.2988452>
- [50] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2018. On the Robustness and Discriminative Power of Information Retrieval Metrics for Top-N Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) (*RecSys '18*). Association for Computing Machinery, New York, NY, USA, 260–268. <https://doi.org/10.1145/3240323.3240347>
- [51] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 109–118.
- [52] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. 2006. Unifying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, Washington, USA) (*SIGIR '06*). Association for Computing Machinery, New York, NY, USA, 501–508. <https://doi.org/10.1145/1148170.1148257>
- [53] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Defu Lian. 2019. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864* (2019).
- [54] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).
- [55] Heitor Werneck, Nicollas Silva, Matheus Carvalho, Adriano C.M. Pereira, Fernando Mourão, and Leonardo Rocha. 2021. A systematic mapping on POI recommendation: Directions, contributions and limitations of recent studies. *Information Systems* (2021), 101789. <https://doi.org/10.1016/j.is.2021.101789>
- [56] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, San Francisco, CA, USA, February 22–25, 2016, Paul N. Bennett, Vanja Josifovski, Jennifer Neville, and Filip Radlinski (Eds.). ACM, 153–162. <https://doi.org/10.1145/2835776.2835837>
- [57] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) (*WSDM '19*). Association for Computing Machinery, New York, NY, USA, 582–590. <https://doi.org/10.1145/3289600.3290975>
- [58] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System. *Comput. Surveys* 52, 1 (Feb 2019), 1–38. <https://doi.org/10.1145/3285029>
- [59] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Y. Li, and Yongdong Zhang. 2020. How to Retrain Recommender System?: A Sequential Meta-Learning Method. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).
- [60] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. *arXiv preprint arXiv:2011.01731* (2020).