Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms

Wayne Xin Zhao[1,2], Junhua Chen[3], Pengfei Wang[4*], Qi Gu[3] and Ji-Rong Wen[1,2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]School of Information, Renmin University of China
[4] Beijing University of Posts and Telecommunications
{batmanfly, cjh1507, guqi@ruc.edu.cn, jrwen}@ruc.edu.cn, wangpengfei@bupt.edu.cn

ABSTRACT

Top-N item recommendation has been a widely studied task from implicit feedback. Although much progress has been made with neural methods, there is increasing concern on appropriate evaluation of recommendation algorithms. In this paper, we revisit alternative experimental settings for evaluating top-N recommendation algorithms, considering three important factors, namely *dataset splitting*, *sampled metrics* and *domain selection*. We select eight representative recommendation algorithms (covering both traditional and neural methods) and construct extensive experiments on a very large dataset. By carefully revisiting different options, we make several important findings on the three factors, which directly provide useful suggestions on how to appropriately set up the experiments for top-N item recommendation.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Top-N Item Recommendation; Evaluation; Experimental Settings

1 INTRODUCTION

In the past decade, top-N item recommendation has been a widely studied task from implicit feedback [1], which aims to identify a small set of items that a user may prefer from a large collection. Various top-N recommendation algorithms have been developed, specially the great progress made with deep learning [2].

To prove the effectiveness of a recommendation algorithm, one needs to construct reliable evaluation experiments on benchmark datasets. Typically, such an evaluation procedure consists of a series of setup steps on datasets, metrics, baselines and other protocols. As each setup step can be conducted with different options, it is essential to develop and design appropriate criterions for standardizing the experimental settings [3, 4].

In the literature, a number of studies have been proposed to standardize evaluation criterions for top-N item recommendation [3–6]. However, they mainly adopt traditional recommendation algorithms as evaluation targets. It is not clear whether some specific finding still holds when neural algorithms are involved in evaluation. As another limit, prior studies may not well respond to recent concerns [7] about evaluation protocols on neural recommendation algorithms. The studied or compared settings in [3–6] do not align with the major divergence from current debate. Besides, existing studies usually use very few comparison methods or datasets. Therefore, there is a need to thoroughly revisit experimental settings of substantial divergence in recent literature, considering both traditional and neural methods.

In this paper, we present a large-scale empirical study on the effect of different experimental settings for evaluating top-N item recommendation algorithms. We try to identify important evaluation settings that have led to major divergence in recent progress [2, 7]. In specific, we consider three important influencing factors, namely dataset splitting, sampled metrics and domain selection. *Dataset splitting* refers to the strategy to construct training, validation and test sets using original data; *sampled metrics* refers to the strategy to compute the metric results with sampled irrelevant items; and *domain selection* refers to the strategy to select suitable datasets from different domains for evaluation.

To examine the effect of the three factors, we construct extensive experiments on the Amazon review dataset [8], containing 142.8 million user-item interaction records from 24 domains. Different from prior works [3, 6], which analyze how each individual method performs under different settings, we study how one selected factor affects the overall performance ranking of different comparison methods, since top-N item recommendation is essentially a ranking task. We select eight representative recommendation algorithms as comparison methods, including both traditional and neural methods. We utilize three ranking correlation measures to quantitatively characterize such ranking differences.

Our empirical study has lead to the following findings. First, for dataset splitting, temporal ordering seems to yield a substantially different performance ranking compared with random ordering. An appropriate option should depend on the specific task. A suggestion is to adopt random ordering in a general setting while temporal ordering for time-sensitive cases (*e.g.,* sequential recommendation). Interestingly, the simple yet widely adopted leave-one-out splitting strategy has a significant correlation with ratio-based splitting strategy. It can be used for small datasets. Second, the performance ranking based on sampled metrics has a relatively weak correlation

*Corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412095

with the exact ranking, and increasing the number of sampled items will improve the correlation degree. When using sampled metrics, researchers should use a large number of sampled irrelevant items as possible. Finally, data domains with varying domain characteristics or sparsity levels are likely to yield substantially different performance rankings. A good strategy is to select representative datasets that are able to cover different aspects of multiple domains.

2 EXPERIMENTAL SETUP

In this section, we set up the experiments by describing the datasets, comparison methods, and evaluation metrics.

Datasets. We adopt the Amazon product review dataset [8] for evaluation, containing 142.8 million reviews from 24 domains. For top-N recommendation, each review is considered as an *interaction record* between a user and an item, while the rest information is discarded, *e.g.,* text and metadata. Since several comparison methods cannot obtain a result in a reasonable time on the largest *book* domain, we remove this domain for the efficiency issue. User-item interaction data from the rest 23 domains as the final dataset. We further adopt the released *five-core copies* of the original review dataset to remove inactive users or infrequent items.

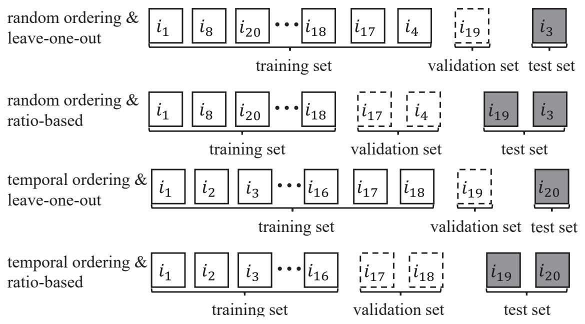
Comparison Methods. We adopt eight recommendation algorithms, including popularity, ItemKNN, SVD++ [9] and BPR [1], DSSM [10], NCF [11], DIN [12] and GCMC [13]. Among these eight methods, popularity and ItemKNN are mainly on based simple global or item-specific statistics, SVD++ and BPR utilize matrix factorization techniques, DSSM and NCF characterize user-item interactions by using neural networks, DIN learns user preference by attending to existing behaviors, and GCMC adopts graph neural networks for recommendation. The eight methods have a good coverage of traditional and neural approaches. We adopt either original or official implementations for these methods. In this paper, we only consider general item recommendation instead of other tasks such as context-aware and sequential recommendation. Note that our focus is not to identify the best algorithm, but study how different experimental settings affect the final performance rankings.

Evaluation Metrics. Top-N item recommendation can be considered as a ranking task, in which the recommendations at top positions are important to consider. Following [4, 14], we use four metrics in following experiments: (1) truncated Precision and Recall at top K positions (P@K and R@K), (2) Mean Average Precision (MAP), and (3) Area under the ROC curve (AUC). We also computed the results for another two metrics of nDCG@K and MRR. They yield the similar results with the above four metrics and omitted.

3 EXPERIMENTAL PROTOCOL

In this section, we present the experimental protocol for comparing experimental settings for top-N item recommendation.

Configuration. We introduce the term "*configuration*" to denote a kind of combination for different options of the three factors, namely *dataset splitting*, *sampled metrics* and *domain selection*. We select the three factors because there is still substantial divergence (lacking standardized discussion) in recent literature of neural methods. Note that we would not enumerate all possible options for the



random ordering & leave-one-out
i_1 i_8 i_{20} ••• i_{18} i_{17} i_4 i_{19} i_3
training set
validation set
test set
random ordering & ratio-based
i_1 i_8 i_{20} ••• i_{18} i_{17} i_4 i_{19} i_3
training set
validation set
test set
temporal ordering & leave-one-out
i_1 i_2 i_3 ••• i_{16} i_{17} i_{18} i_{19} i_{20}
training set
validation set
test set
temporal ordering & ratio-based
i_1 i_2 i_3 ••• i_{16} i_{17} i_{18} i_{19} i_{20}
training set
validation set
test set

Figure 1: An illustrative example for four splitting strategies on a sample user. The user has interacted with twenty items. The subscript of an item i denotes the interaction order with the user: a smaller index indicates an earlier interaction time. We use normal, dash-lined and grey boxes to denote the training, validation and test sets, respectively.

three factors. Instead, we only consider popular or controversial options from recent studies [2]. In order to reduce the influence of other factors, we either report results separately by different options or set them to the suggested option by prior studies [3–6].

Correlation Measurement. Given a configuration, we can obtain a ranked list of the eight comparison methods, called *performance ranking*, according to the descending order of their performance based on some metric. We adopt three measures to quantitatively characterize the correlation or similarity degree between two performance rankings: (1) *Overlap Ratio at top-K positions* (OR@k) computing the overlap ratio of top k methods between two ranked lists; (2) *Spearman's Rank Correlation* (SRC) measuring the association between two ranked lists; and (3) *Inversion Pair Count* (IPC) counting the number of inversion pairs between two ranked lists. The reason to select the three measures is given as follows. First, correlation measures (*e.g.,* Spearman or Kendall) are commonly used to reflect ranking differences in prior evaluation studies [6]. Second, IPC provides a more intuitive understanding on the values of SRC. Finally, for item recommendation, top positions are more important to consider, which is captured by OR@k.

Procedure Overview. Given a metric, we first derive a performance ranking of the eight methods according to some configuration (optimized with validation sets). To examine the effect of one factor, we will accordingly generate multiple configurations by considering the alternative options. Then, we compute the correlation degree between the performance rankings under two different configurations using the above measures. Finally, the correlation results will be averaged over 23 data domains (except Section 4.3).

4 EXPERIMENT

In this section, we present the experiment results related to the three factors, namely *dataset splitting*, *sampled metrics* and *domain selection*. When considering one factor, we fix the rest two factors. That is to say, given two configurations to compare, we only vary the studied factor, while the rest settings will be set to the same in two compared configurations.

4.1 Analysis on Dataset Splitting

We first study the effect of different dataset splitting strategies (*i.e.,* constructing training/validation/test sets) on performance ranking.

Setting. For each user, we first organize the interaction records of a user using two methods: (1) *Random Ordering (RO)* randomly shuffles the items; (2) *Temporal Ordering (TO)* sorts the items according to their interaction timestamps. Then, the reordered user-item interaction sequences can be split using two common methods: (1) *Ratio-based Splitting (RS)* splits the dataset into three parts for training, validation and test according to a predefined ratio, which is set to 8:1:1 here. We repeat the process for five times to generate different evaluation sets for computing average results. (2) *Leave-one-out Splitting (LS)* selects one ground-truth item as test set and another one as validation set, while the rest items are considered as training set. *LS* strategy can be considered as a special case for *RS*, where both validation and test sets contain only one item. Considering both ordering and splitting, we generate four combinations in total, which are illustrated in Figure 1.

Results. We present the comparison results between two different configurations in Table 1. First, compared with the splitting method (either *ratio* or *leave-one-out*), the item ordering way (either *random* or *temporal*) seems to have a more significant influence on the performance ranking. For each metric, the correlation values from the first two lines are substantially weaker than the last two lines. With temporal ordering, it is essentially similar to the setting of sequential recommendation. The results indicate that the option of item ordering should depend on specific recommendation tasks. It is suggested to adopt *random ordering* in a general setting (especially for evaluating time-insensitive recommendation algorithms), while adopt *temporal ordering* in time-sensitive cases (*e.g.,* sequential recommendation). Second, with the same item ordering way, the two splitting methods yield very similar ranked lists (see the last two lines for each metric). Indeed, leave-one-out splitting has been widely adopted in recent literature [8, 11]. We suggest using ratio-based splitting when possible for more accurate evaluation. While, leave-one-out splitting seems to be preferred with small datasets, since we can use more training data for alleviating data sparsity.

4.2 Analysis on Sampled Metrics

Next, we study the effect of sampled metrics (*i.e.,* only a smaller set of sampled items and the ground-truth items are ranked for computing the metrics) on performance ranking.

Setting. For test, it is time-consuming to take all the items from the item set as the candidate when the size of item set is large. An alternative way is to sample a small set of items as irrelevant items. Then, the ground-truth items and sampled items are merged as a single candidate list for ranking. The results of the metrics will be computed based on such an item subset. This way is called *sampled metrics* [7]. We consider two sampling strategies, namely *uniform sampling* and *popularity-biased sampling*, which samples irrelevant items according to either a uniform distribution or a frequency-based distribution. We further consider using three different numbers of irrelevant samples, namely $\{10, 50, 100\}$, which means that a ground-truth item will be paired with 10, 50 or 100 sampled items. When we adopt leave-one-out splitting, the case becomes *real-plus-N* [3, 6]. For comparison, we adopt the entire item set (excluding ground-truth items) for ranking (denoted by *all*) as a referencing setting. Following Section 4.1, for dataset splitting, we adopt ratio-based dataset splitting (denoted by *RS*) with random ordering (denoted by *RO*) in all compared configurations.

Table 1: Correlation comparison with different configurations on dataset splitting. The results are averaged over the 23 domains (with standard deviations). Here, "RS" and "LS" denote ratio-based or leave-one-out splitting; "RO" and "TO" denote random or temporal ordering. "↑" ("↓") indicates a larger (smaller) result is better.

Metrics	Comparison	OR@3 (↑)	SRC (↑)	IPC (↓)
P@10	(TO, RS) *v.s.* (RO, RS)	$0.731_{\pm 0.232}$	$0.687_{\pm 0.221}$	$3.588_{\pm 2.042}$
	(TO, LS) *v.s.* (RO, LS)	$0.757_{\pm 0.198}$	$0.709_{\pm 0.237}$	$2.826_{\pm 2.306}$
	(RO, LS) *v.s.* (RO, RS)	$0.916_{\pm 0.144}$	$0.875_{\pm 0.155}$	$1.212_{\pm 0.994}$
	(TO, LS) *v.s.* (TO, RS)	$0.815_{\pm 0.199}$	$0.755_{\pm 0.184}$	$1.660_{\pm 1.762}$
R@10	(TO, RS) *v.s.* (RO, RS)	$0.727_{\pm 0.266}$	$0.705_{\pm 0.274}$	$3.499_{\pm 2.418}$
	(TO, LS) *v.s.* (RO, LS)	$0.698_{\pm 0.221}$	$0.649_{\pm 0.243}$	$3.866_{\pm 2.557}$
	(RO, LS) *v.s.* (RO, RS)	$0.901_{\pm 0.152}$	$0.872_{\pm 0.144}$	$1.650_{\pm 1.464}$
	(TO, LS) *v.s.* (TO, RS)	$0.828_{\pm 0.190}$	$0.769_{\pm 0.242}$	$2.039_{\pm 1.282}$
AUC	(TO, RS) *v.s.* (RO, RS)	$0.727_{\pm 0.195}$	$0.702_{\pm 0.190}$	$3.168_{\pm 2.281}$
	(TO, LS) *v.s.* (RO, LS)	$0.797_{\pm 0.244}$	$0.715_{\pm 0.235}$	$2.644_{\pm 2.299}$
	(RO, LS) *v.s.* (RO, RS)	$0.915_{\pm 0.147}$	$0.825_{\pm 0.277}$	$1.431_{\pm 1.124}$
	(TO, LS) *v.s.* (TO, RS)	$0.905_{\pm 0.151}$	$0.823_{\pm 0.199}$	$1.170_{\pm 1.148}$
MAP	(TO, RS) *v.s.* (RO, RS)	$0.769_{\pm 0.234}$	$0.762_{\pm 0.265}$	$2.909_{\pm 2.337}$
	(TO, LS) *v.s.* (RO, LS)	$0.757_{\pm 0.239}$	$0.696_{\pm 0.252}$	$3.212_{\pm 2.677}$
	(RO, LS) *v.s.* (RO, RS)	$0.930_{\pm 0.135}$	$0.911_{\pm 0.134}$	$1.130_{\pm 1.288}$
	(TO, LS) *v.s.* (TO, RS)	$0.884_{\pm 0.182}$	$0.799_{\pm 0.172}$	$2.088_{\pm 1.900}$

Results. Table 2 presents the correlation results of different sampled metrics, which are much smaller than those in Table 1. It indicates that using sampled metrics has a larger influence on performance ranking. Indeed, such a concern has been discussed in recent studies [7]: sampled metrics are likely to be inconsistent and do not even persist qualitative ordering. Another observation is that sampling more irrelevant items increases the correlation degree between the sampled and exact metrics. Finally, different sampling strategies are likely to cause the performance turbulence of some specific algorithms, which substantially affects the performance ranking. Comparing two sampling strategies, it seems that uniform sampling is more closely correlated with the entire ranking. Generally, sampled metrics should not be used for small datasets. If needed, we suggest using more irrelative items (*e.g.,* 1000 items suggested by [9]).

4.3 Analysis on Domain Selection

Above, we compute the correlation results by averaging over 23 different domains. Here, we consider whether different domains lead to varying performance rankings. Such an issue is useful to answer how to select suitable datasets for evaluation.

Setting. Given two domains, we first generate a configuration according to the suggested setting, *i.e.,* (RO, RS, all), from Section 4.1 and 4.2, and then obtain a performance ranking for each domain under the configuration for some metric. Then, we compute the *Spearman's Rank Correlation* score between two domain-specific

Table 2: Correlation comparison with different configurations on sampled metrics. The results are averaged over 23 domains. "$\Delta = \{RO, RS\}$" denotes random ordering and ratio-based splitting method for dataset splitting, "pop" / "uni" denote popularity/uniform sampling, the subscript denotes the number of sampled items and "all" denotes all non-ground-truth items.

Metric	Comparison	OR@3 (↑)	SRC (↑)	IPC (↓)
P@10	(Δ, all) v.s. (Δ, pop_{10})	$0.531_{\pm 0.302}$	$0.566_{\pm 0.285}$	$6.111_{\pm 3.971}$
	(Δ, all) v.s. (Δ, pop_{50})	$0.577_{\pm 0.297}$	$0.604_{\pm 0.300}$	$5.799_{\pm 3.289}$
	(Δ, all) v.s. (Δ, pop_{100})	$0.709_{\pm 0.207}$	$0.606_{\pm 0.328}$	$5.223_{\pm 2.168}$
	(Δ, all) v.s. (Δ, uni_{10})	$0.678_{\pm 0.349}$	$0.667_{\pm 0.329}$	$3.644_{\pm 2.801}$
	(Δ, all) v.s. (Δ, uni_{50})	$0.777_{\pm 0.229}$	$0.722_{\pm 0.254}$	$2.966_{\pm 2.424}$
	(Δ, all) v.s. (Δ, uni_{100})	$0.872_{\pm 0.189}$	$0.804_{\pm 0.178}$	$2.357_{\pm 1.200}$
R@10	(Δ, all) v.s. (Δ, pop_{10})	$0.581_{\pm 0.305}$	$0.544_{\pm 0.292}$	$6.670_{\pm 2.949}$
	(Δ, all) v.s. (Δ, pop_{50})	$0.667_{\pm 0.231}$	$0.607_{\pm 0.273}$	$5.711_{\pm 2.302}$
	(Δ, all) v.s. (Δ, pop_{100})	$0.712_{\pm 0.200}$	$0.619_{\pm 0.318}$	$5.143_{\pm 2.166}$
	(Δ, all) v.s. (Δ, uni_{10})	$0.801_{\pm 0.161}$	$0.695_{\pm 0.212}$	$2.579_{\pm 1.672}$
	(Δ, all) v.s. (Δ, uni_{50})	$0.864_{\pm 0.194}$	$0.788_{\pm 0.177}$	$1.891_{\pm 1.621}$
	(Δ, all) v.s. (Δ, uni_{100})	$0.872_{\pm 0.189}$	$0.804_{\pm 0.173}$	$2.164_{\pm 1.215}$
AUC	(Δ, all) v.s. (Δ, pop_{10})	$0.659_{\pm 0.272}$	$0.690_{\pm 0.233}$	$5.379_{\pm 2.852}$
	(Δ, all) v.s. (Δ, pop_{50})	$0.673_{\pm 0.281}$	$0.726_{\pm 0.205}$	$5.066_{\pm 2.399}$
	(Δ, all) v.s. (Δ, pop_{100})	$0.695_{\pm 0.278}$	$0.737_{\pm 0.197}$	$4.922_{\pm 2.311}$
	(Δ, all) v.s. (Δ, uni_{10})	$0.844_{\pm 0.224}$	$0.764_{\pm 0.253}$	$1.994_{\pm 2.295}$
	(Δ, all) v.s. (Δ, uni_{50})	$0.868_{\pm 0.210}$	$0.793_{\pm 0.247}$	$1.893_{\pm 2.290}$
	(Δ, all) v.s. (Δ, uni_{100})	$0.878_{\pm 0.181}$	$0.813_{\pm 0.230}$	$1.629_{\pm 1.644}$
MAP	(Δ, all) v.s. (Δ, pop_{10})	$0.599_{\pm 0.315}$	$0.559_{\pm 0.288}$	$5.780_{\pm 2.159}$
	(Δ, all) v.s. (Δ, pop_{50})	$0.610_{\pm 0.298}$	$0.588_{\pm 0.241}$	$5.212_{\pm 1.987}$
	(Δ, all) v.s. (Δ, pop_{100})	$0.664_{\pm 0.299}$	$0.667_{\pm 0.262}$	$4.314_{\pm 1.829}$
	(Δ, all) v.s. (Δ, uni_{10})	$0.772_{\pm 0.206}$	$0.709_{\pm 0.252}$	$2.909_{\pm 1.347}$
	(Δ, all) v.s. (Δ, uni_{50})	$0.830_{\pm 0.166}$	$0.758_{\pm 0.212}$	$2.124_{\pm 1.199}$
	(Δ, all) v.s. (Δ, uni_{100})	$0.838_{\pm 0.164}$	$0.800_{\pm 0.214}$	$1.768_{\pm 1.088}$

rankings. We average the *SRC* scores over all the metrics. The final score is used to measure the correlation between two domains.

Results. Figure 2 presents the average correlation results between two domains. We reorder the rows or columns so that large values can be aggregated in the diagonal line. Interestingly, the entire heatmap seems to contain four major groups, in which within-group correlation values are higher than those across groups. The results indicate that different domains are likely to yield varying performance rankings under the same configuration. Hence, domain difference should be considered in evaluation. By inspecting into the dataset, we find that domain characteristics (*e.g.,* the first group mostly corresponding to digital products) and sparsity levels (*e.g.,* the ratio of user-item interaction) seem to have significant effect on the correlation results. A good strategy is to use several datasets of varying sparsity levels from diverse domains. Here, "domains" refer to the categories of the Amazon dataset. Although these domains come from the same platform, the finding has reflected the concern to some extent when selecting datasets for evaluation. We will examine this issue using more datasets in future work.

5 CONCLUSION

In this paper, we empirically compared different experimental settings of three important factors for top-N item recommendation. Our experiments have led to several empirical suggestions for evaluating item recommendation algorithms. First, for dataset splitting, random ordering with ratio-based splitting is the suggested option

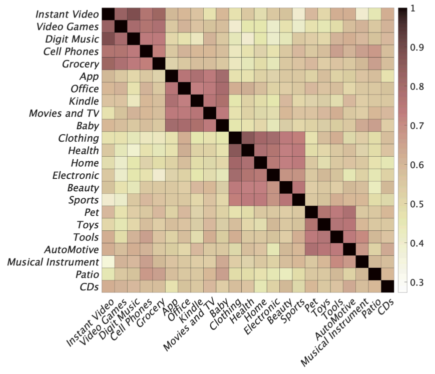


Figure 2: Visualization of pairwise domain correlations. Each cell indicates the computed correlation score between two domains (a darker color indicates a larger value).

for evaluating time-insensitive algorithms, while leave-one-out splitting can be applied to small datasets. Second, we should be careful to use sampled metrics. If it was used, we suggest sampling a large number of items. Third, it is suggested to use multiple datasets from diverse domains as evaluation sets. As future work, we will consider constructing online evaluation for studying the effect of various factors. Also, more factors and datasets will be investigated for evaluating recommendation algorithms.

6 ACKNOWLEDGEMENTS

This research work was partially supported by the National Natural Science Foundation of China under Grant No. 61872369, 61832017 and 61802029, Beijing Academy of Artificial Intelligence (BAAI) under Grant No.BAAI2020ZJ0301, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098.

REFERENCES

[1] S. Rendle, C. Freudenthaler, Z. Gantner, BPR: Bayesian Personalized Ranking from Implicit Feedback, in: UAI 2009, 452–461, 2009.
[2] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep Learning Based Recommender System: A Survey and New Perspectives, ACM Comput. Surv. 52 (1) (2019) 5:1–5:38.
[3] A. Bellogín, P. Castells, I. Cantador, Precision-oriented evaluation of recommender systems: an algorithmic comparison, in: ACM RecSys, 333–336, 2011.
[4] T. Silveira, M. Zhang, X. Lin, Y. Liu, S. Ma, How good your recommender system is? A survey on evaluations in recommendation, JMLC 10 (5) (2019) 813–831.
[5] H. Steck, Evaluation of recommendations: rating-prediction and ranking, in: ACM, RecSys 2013, 213–220, 2013.
[6] A. Said, A. Bellogín, Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks, in: ACM RecSys, 129–136, 2014.
[7] W. Krichene, S. Rendle, On Sampled Metrics for Item Recommendation, in: ACM SIGKDD, 2020.
[8] R. He, J. J. McAuley, Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering, in: WWW 2016, 507–517, 2016.
[9] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: ACM, SIGKDD 2008, 426–434, 2008.
[10] P. Huang, X. He, J. Gao, L. Deng, A. Acero, L. P. Heck, Learning deep structured semantic models for web search using clickthrough data, in: CIKM, 2013.
[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural Collaborative Filtering, in: WWW, 2017, 173–182, 2017.
[12] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, K. Gai, Deep Interest Network for Click-Through Rate Prediction, in: ACM, SIGKDD, 1059–1068, 2018.
[13] R. van den Berg, T. N. Kipf, M. Welling, Graph Convolutional Matrix Completion, CoRR abs/1706.02263.
[14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.