

Off-line vs. On-line Evaluation of Recommender Systems in Small E-commerce

Ladislav Peska

Department of Software Engineering,
Faculty of Mathematics and Physics, Charles University
peska@ksi.mff.cuni.cz

Peter Vojtas

Department of Software Engineering,
Faculty of Mathematics and Physics, Charles University
vojtas@ksi.mff.cuni.cz

ABSTRACT

In this paper, we present our work towards comparing on-line and off-line evaluation metrics in the context of small e-commerce recommender systems. Recommending on small e-commerce enterprises is rather challenging due to the lower volume of interactions and low user loyalty, rarely extending beyond a single session. On the other hand, we usually have to deal with lower volumes of objects, which are easier to discover by users through various browsing/searching GUIs.

The main goal of this paper is to determine applicability of off-line evaluation metrics in learning true usability of recommender systems (evaluated on-line in A/B testing). In total 800 variants of recommending algorithms were evaluated off-line w.r.t. 18 metrics covering rating-based, ranking-based, novelty and diversity evaluation. The off-line results were afterwards compared with on-line evaluation of 12 selected recommender variants and based on the results, we tried to learn and utilize an off-line to on-line results prediction model.

Off-line results shown a great variance in performance w.r.t. different metrics with the Pareto front covering 64% of the approaches. Furthermore, we observed that on-line results are considerably affected by the seniority of users. On-line metrics correlates positively with ranking-based metrics (AUC, MRR, nDCG) for novice users, while too high values of novelty had a negative impact on the on-line results for them.

1 INTRODUCTION

Recommender systems (RS) belong to the class of automated content-processing tools, aiming to provide users with unknown, surprising, yet relevant objects without the necessity of explicitly query for them. The core of recommender systems are machine learning algorithms applied on the matrix of user to object preferences. As such, recommender systems are highly studied research topic as well as extensively used in real-world applications.

However, throughout the decades of recommender systems research, there was a discrepancy between industry and academia in evaluation of proposed recommending models. While academic researchers often focused on off-line evaluation scenarios based on recorded past data, industry practitioners value more the results of on-line experiments on live systems, e.g., via A/B testing. While off-line evaluation is easier to conduct, repeatable, fast and can incorporate arbitrary many recommending models, it is often argued that it does not reflect well the true utility of recommender systems as seen in on-line experiments [7]. On-line evaluation is able to

naturally incorporate current context, tasks or search needs of the user, appropriateness of recommendations' presentation as well as causality of user behavior. On the other hand, A/B testing on live systems is time consuming, the necessary time scales linearly with the volume of evaluated approaches and it can even harm retailer's reputation if bad recommendations are shown to users.

1.1 Bridging the Off-line vs. On-line Gap

A wide range of approaches aimed to bridge the gap between industry and academia. Jannach and Adomavicius [11] argue for recommendations with a purpose, i.e., after a certain level of RS's maturity. In particular, after the numerical estimators of user's preference are established, authors suggest to step back and revisit some of the foundational aspects of RS. Authors aimed to reconsider the variety of purposes, for which recommender systems are already used today in a more systematic manner and proposed a framework which should cover both consumer's/provider's viewpoint and strategic/operational perspective. One way to approach this goal are user studies via questionnaires (e.g. [24]) or more involved frameworks, e.g. [17]. Still, the main problem remains: we may lack the participants, whose motivation, information needs and behavior would be similar to real-world users.

Another approach to treat the off-line/on-line phenomenon comes from considerations about relevance of statistical learning in understanding causation, confounding, missing (not at random - MNAR) data (see e.g., [19]). A starting point of these approaches is the observation that implicit feedback (despite many advantages) has inherent biases and these are key obstacles to its effective usage. For example, position bias in search rankings strongly influences how many clicks a result receives, so that directly using click-through data as a training signal in Learning-to-Rank (LTR) methods yields sub-optimal results [14]. To overcome the bias problem, Joachims et al. [15] presented a counterfactual inference framework that provides the theoretical basis for unbiased LTR via Empirical Risk Minimization despite the biased data. Also Gilotte et al. [7] utilized de-biased off-line methods to estimate the potential uplift of the on-line performance. Authors proposed a new counterfactual estimator and evaluated it on a proprietary dataset of 39 past A/B tests, containing several hundreds of billions of recommendations in total.

A recent contribution to academia-industry discussion was the 2017 Recommender Systems Challenge [1], focused to the problem of job recommendations¹. In the first phase, participants evolved their models on off-line data. Afterwards, invited participants were tasked to provide and evaluate recommendations on-line. Most of the teams managed to preserve their off-line performance also

This is an author version submitted to ACM Hypertext 2020, ,
2020. ACM ISBN 978-1-4503-7098-1/20/07...\$15.00
<https://doi.org/10.1145/3372923.3404781>

¹<http://www.recsyschallenge.com/2017/>

during the on-line phase. Quite surprising was the fact that traditional methods and metrics to estimate the users' preferences for unknown items (of course, tuned to specifics of the task) worked best. The winning team combined content and neighbor-based models with feature extraction, balanced sampling and minimizing a tricky classification objective [27].

1.2 Recommender Systems in Small E-commerce

Previously mentioned approaches were mostly user-centric. However, in the RecSys Challenge 2017 [1], we could observe the success of item-based methods. The main cause was probably the cold start problem, which is prevalent also in small e-commerce enterprises. Kaminskas et al. [16] observed that the small amount of returning customers makes traditional user-centric personalization techniques inapplicable and designed an item-centric product recommendation strategy. Authors deployed the proposed solution on two retailers' websites and evaluated it in both on-line and off-line settings. Jannach et al. [12] considered the problem of recommending to users with short-term shopping goals. Authors observed the necessity of item-based approaches but also importance of algorithms usually used for long-term preferences.

Peska and Vojtas [23] proposed the usage of implicit preferences relations on the problem of recommending for small e-commerce enterprises with short-term user's goals. Their work is based on an complex observation of users' behavior up to the level of noticeability of individual objects on the category pages.

In general, providing recommendation service on small e-commerce enterprises brings several specific challenges and opportunities, which changes some recommending paradigms applied, e.g., in large-scale multimedia enterprises. Let us briefly list the key challenges:

- High competition has a negative impact on user loyalty. Typical sessions are very short, users quickly leave to other vendors, if their early experience is not satisfactory enough. Only a fraction of users ever returns.
- For those single-time visitors, it is not sensible to provide any unnecessary information (e.g., ratings, reviews, registration details).
- Consumption rate is low, users often visit only a handful (0-5) of objects and rarely ever buys anything.
- Small e-commerce enterprises generally offer lower volume of objects (ranging usually from hundreds to tens of thousands instead of millions as in, e.g., Amazon).
- Objects often contain extensive textual description as well as a range of categorical attributes. Browsing and attribute search GUIs are present and widely used.

The first three mentioned factors contribute to the data sparsity problem and limit applicability of user-based collaborative filtering (CF). Although the total number of users may be relatively large (hundreds or thousands per day), the volume of visited objects per user is limited and the timespan between the first and last feedback is short. The last two factors contribute towards objects' discoverability. This may seemingly decrease the necessity of recommender

systems², but also decreases the effect of missing not at random data [21] and therefore may contribute to the consistency of off-line and on-line evaluation. Also, in many product domains (including our travel agency test bed), it is uncommon to have any "well-known" items, such as blockbuster movies or popular songs. This further limits applicability of counterfactual approaches.

Despite mentioned obstacles, the potential benefit of recommender systems in small e-commerce enterprises is still considerable, e.g., "more-of-the-kind" and "related-to-purchased" recommendations are not easy to mimic with standard search/browsing GUI.

1.3 Off-line to On-line Predictions

Garcin et al. [6] focused on news recommendations and observed a major difference between off-line and on-line accuracy evaluations. These differences went beyond a small numerical variance and had a determining impact on the ordering of best methods. Utilized metric (hit@top-3) is somewhat proprietary, but supported by the website design. In a follow-up study [20], authors focused on additional off-line metrics including accuracy, diversity, coverage and serendipity metrics. Similarly as in [20], we usually observed very high correlation scores for metrics from a single cluster (ranking-prediction, rating-prediction), but correlations between metrics from different clusters were much lower in our case. This work also inspired us to employ regularized linear regression models to predict on-line performance from off-line results.

Rossetti et al. [26] focused on the MovieLens dataset and organized a user study aiming to compare off-line and on-line evaluation metrics. Authors specifically distinguished long-tail recommendations and recommendations of previously unknown items. Similarly as in [6], same metrics were evaluated off-line and on-line. Authors showed that off-line evaluation induces similar ranking of algorithms, but with some exceptions. Also Beel et al. [3] focused on ranking accuracy metrics such as nDCG and MRR in a literature RS. Authors reported on some moderate correlations between CTR and these off-line metrics, but also mentioned several cases, where the prediction failed. Gruson et al. [8] focused on the problem of candidates selection for on-line evaluation in Spotify playlists recommendations. Authors employed several approaches to de-bias the off-line evaluations based on importance sampling, where some approaches have seemingly good prediction results. However, authors did not compare these models with original "biased" feedback. Therefore, it is hard to assess the importance of feedback de-biasing. As the nature of small e-commerce domains seemingly reduces such feedback biases, we did not include such approaches in our current study, but we plan to explore them in the future work. Let us also note that none of [3, 8, 26] considered other off-line metrics than some form of ranking accuracy.

Although the mentioned related studies (as well as our own work) share the general goal of observing and describing relations between off-line and on-line results, there is a determining difference in the considered application domains. This has an effect on both the choice of recommending algorithms, evaluation metrics

² Although objects are more discoverable and users do not depend on recommendations only, they are often not willing to spend too much time in the discovery process and recommendations may considerably shorten it.

as well as the generic study design. Specifically, we are not aware of any work considering the predictability of on-line results from off-line metrics in the context of small e-commerce enterprises. We also evaluated a wide range of off-line metrics beyond ranking accuracy and evaluated the effect of promoting diversity or novelty of recommendations. We would also like to note that mentioned papers only considered a single class of recommending algorithms, while in our work, we evaluated several diverse recommending algorithms.

1.4 Main Contributions

The main scope of this paper is to contribute towards bridging the gap between industry and academia in the evaluation of recommender systems. We specifically focused on the domain of small e-commerce enterprises and within this scope, we aim on determining the usability of various off-line evaluation methods and their combinations in learning the relevance of recommendations w.r.t. on-line production settings. In total, 800 variants of recommender systems (3 base recommending algorithms combined with 9 user profile construction algorithms and various hyperparameter settings) were evaluated off-line w.r.t. 18 metrics covering rating-based, ranking-based, novelty and diversity metrics. The off-line results were afterwards compared with on-line evaluation of 12 selected algorithm's variants.

To sum up, main contributions of this paper are as follows:

- We compared a wide range of off-line metrics against the actual on-line results w.r.t. click through rate (CTR) and visits after recommendation (VRR).
- The observed results highly depend on users "seniority". While, the ranking-based metrics generally correlate with on-line results for less senior users, novelty and diversity gain importance for users with more visited objects.
- We further evaluated several simple regression techniques aiming to predict on-line results based on the off-line metrics and achieved considerable predictability of CTR and VRR under leave-one-out cross-validation (LOOCV) scenario.
- Based on the previous point, we may recommend word2vec and some variants of cosine CB methods to be used on small e-commerce enterprises.
- Datasets acquired during both off-line and on-line evaluation are available for future work.

2 MATERIALS AND METHODS

2.1 Dataset and Evaluation Domain

As the choice of suitable recommending algorithms is data-dependent, let us first briefly describe the dataset and the domain, we used for evaluation.

Experiments described in this paper were conducted on a medium-sized Czech travel agency. The agency sells tours of various types to several dozens of countries. Each object (tour) is available in selected dates. Some tours (such as trips to major sport events) are one-time only events, others, e.g., seaside holidays or sight-seeing tours are offered on a similar schedule with only minimal changes for several years. All tours contain a textual description accompanied with a range of content-based (CB) attributes, e.g.,

tour type, meal plan, type of accommodation, length of stay, prices, destination country/ies, points of interest etc.

The agency's website contains simple attribute and keyword search GUI as well as extensive browsing and sorting options. Recommendations are displayed on a main page, browsed categories, search results and opened tours. However, due to the importance of other GUI elements, recommendations are usually placed below the initially visible content.

2.2 Recommending Algorithms

In accordance with Kaminskas et al. [16], we considered user-based recommending algorithms, e.g., matrix factorization models impractical for small e-commerce due to a high user fluctuation and short timespan between user's first and last visits.

2.2.1 Item-to-item Recommending Models. We considered three recommending approaches corresponding with the three principal sources of data: object's CB attributes, their textual description and the history of users' visits (collaborative filtering). The information sources are mostly orthogonal, each focused on a different recommending paradigm. The expected output of recommendations based on CB attributes is to reveal similar objects to the ones in question. By utilizing the stream of user's visits, it is possible to uncover objects that are related, yet not necessarily similar. The expected outcome of textual-based approaches is also to provide similar objects, however the similarity may be hidden within the text, e.g., seaside tours with the same type of beach, both suitable for families, located in a small peaceful village, but in a different country. For each type of source information, we proposed one recommending algorithm as follows:

– Skip-gram **word2vec** model [22] utilizes the stream of user's visits. Similarly as in [2], the sequence of visited objects is used instead of a sentence of words, however, we kept the original window size parameter in order to better model the stream of visits. The output of the algorithm is an embedding of a given size for each object, while similar embeddings denotes objects appearing in a similar context. In evaluation, embedding's size was selected from {32, 64, 128} and context window size was selected from {1, 3, 5}.

– **Doc2vec** model [18] utilizes the textual description of objects. Doc2vec extends word2vec model by an additional attribute defining the source document (object) for each word in question. The model, in addition to the word embeddings calculates also embeddings of the document itself, therefore the output of the algorithm are embeddings of a given size for each object (document). Textual descriptions of objects were preprocessed by a stemmer³ and stop-words removal. In evaluation, embedding's size was selected from {32, 64, 128} and window size from {1, 3, 5}.

– Finally, we used **cosine similarity** on CB attributes. Nominal attributes were binarized, while numeric attributes were standardized before the similarity calculation. We evaluated two variants of the approach differing in whether to allow evaluating similarity on self⁴. In this way, we may promote/restrict recommendations of already visited objects, which belongs to some of the commonly used strategies.

³Language and link removed for the sake of anonymization

⁴Otherwise, the similarity of an object to itself is zero by definition.

Given a query of a single object, the base recommended list would be a list of top- k objects most similar to the query object (or its embeddings vector).

2.2.2 Using History of User's Visits. While the above described algorithms focus on modeling item-item relations, we may possess a longer record of visited objects for some users. Although many approaches focused on a last visited object only, e.g., [16], some approaches using the whole user session emerged recently [10].

Therefore, we proposed in total nine methods to process users' history and aggregate recommendations for individual objects. The variants are as follows:

- Using **mean** of recommendations for all visited objects.
- For each candidate object, use **max** of its similarity w.r.t. some of the visited object.
- Using **last** visited object only.
- Using weighted average of recommendations with linearly decreasing weights. In this case, only the **last- k** visited objects are considered, while its weight $w = 1 - (\text{rank}/k)$ linearly decreases for older visits. We evaluated results considering last 3, 5 and 10 objects.
- Using weighted average of recommendations with **temporal** weights. This variant is the same as the previous one, except that the weights of objects are calculated based on the timespan between the current date and the date of visit: $w = 1/(\log(\text{timespan.days}) + \epsilon)$. We evaluated results considering last 3, 5 and 10 objects as well as a full user profile.

While the first two approaches considered uniform importance of the visited objects, others rely on some variations of "*the newer the better*" heuristic. Using history of the user instead of the last item only is one of the extensions of our work compared to [16].

2.2.3 Novelty and Diversity Enhancements. The performance of recommenders may also depend on a lot of subjective, user-perceived criteria, as introduced in [25], such as *novelty* or *diversity* of recommended items. Therefore, in the off-line evaluation (Section 3.1, we evaluated one type of diversity metric (intra-list diversity [5]) and two types of novelty metrics (temporal novelty considering the timespan from the last object's update and user-perceived novelty describing the fraction of recommended objects, which were previously visited by the user).

However, as certain types of algorithms may provide recommendations that lack sufficient novelty or diversity, we also utilized strategies enhancing temporal novelty and diversity. Both novelty and diversity enhancements were applied as a post-processing of the lists of recommended objects. For diversity enhancements, we adopted the Maximal Margin Relevance approach [4] with λ parameter held constant at 0.8 and item-to-item similarity defined as a cosine similarity of their CB attributes. For enhancing temporal novelty, we re-ranked the list of recommended objects based on a weighted average of their original relevance r and temporal novelty novelty_t :

$$\bar{r}_o = \lambda * r_o + (1 - \lambda) * \text{novelty}_t(o) \quad (1)$$

Novelty_t applies a logarithmic penalty on the time passed from the last object's update (see Eq. 2). The λ parameter was held constant at 0.8.

As the choice of a recommending algorithm, user's history aggregation, novelty and diversity enhancements are orthogonal, we run the off-line evaluation for all possible combinations. In total, 800 variants of RS were evaluated.

3 EVALUATION SCENARIO

In this section, we would like to describe the evaluation scenario and metrics. We separate the evaluation into two distinct parts: off-line evaluation on historic data and on-line A/B testing on a production server.

3.1 Off-line Evaluation

For the off-line experiment, we recorded users' visits for the period of two and half years. The dataset contained over 560K records from 370K users. However, after applying restrictions on the volume of visits⁵, the resulting dataset contained 260K records of 72K users. We split the dataset into a train set and a test set based on a fixed time-point, where the interactions collected during the last month and half were used as a test set. The test set was further restricted to only incorporate users, who have at least one record in the train set as well, resulting into 3400 records of 970 users.

In evaluation, we focused on four types of metrics, commonly used in recommender system's evaluation: rating prediction, ranking prediction, novelty and diversity. We evaluated several metrics for each class.

For rating prediction, we suppose that visited objects have the rating $r = 1$ and all others $r = 0$. Mean absolute error (MAE) and coefficient of determination (R^2) were evaluated.

For ranking-based metrics, we supposed that the relevance of all visited objects is equal, $r = 1$ and other objects are irrelevant, $r = 0$. Following metrics were evaluated: area under ROC curve (AUC), mean average precision (MAP), mean reciprocal rank (MRR), precision and recall at top-5 and top-10 recommendations (p5, p10, r5, r10) and normalized discounted cumulative gain at top-10, top-100 and a full list of recommendations (nDCG10, nDCG100, nDCG). The choice of ranking metrics reflects the importance of the head of the recommended list (p5, p10, r5, r10, nDCG10, MRR, MAP) as only a short list of recommendations can be displayed to the user. However, as the list of recommendable objects may be restricted due to the current context of the user (e.g., currently browsed category), we also included metrics evaluating longer portions of the recommended lists (AUC, nDCG100, nDCG).

As discussed in section 2.2.3, we distinguish two types of novelty in recommendations: recommending recently created or updated objects (temporal novelty) and recommending objects not seen by the user in the past (user novelty). For temporal novelty, we utilized logarithmic penalty on the timespan between current date and the date of the object's last update:

$$\text{novelty}_t = 1/(\log(\text{timespan.days}) + \epsilon) \quad (2)$$

Mean of novelty_t for top-5 and top-10 recommendations was evaluated. For user novelty, a fraction of already known vs. all recommended objects was used: $\text{novelty}_u = 1 - |o \in \text{top-}k \cap o \in \text{known}_u|/k$ and evaluated for top-5 and top-10 objects. Finally, the

⁵Only the users with at least 2 and no more than 150 visited objects were kept.

intra-list diversity (ILD) [5] evaluated at top-5 and top-10 recommendations was utilized as a diversity metric.

All off-line metrics were evaluated for each pair of user and recommender. Mean values for each recommender are reported.

3.2 On-line Evaluation

The on-line A/B testing was conducted on the travel agency's production server during the period of one month. Out of 800 RS variants evaluated off-line, we selected in total 12 recommenders with (close to) best and (close to) worst results w.r.t. each evaluated metric. Details of the selection procedure are in section 4.1. One recommender was assigned to each user, based on his/her ID (i.e. *UID%12*). During the on-line evaluation, we monitored which objects were recommended to the user, whether (s)he clicked on some of them and which objects (s)he visited. The website tracks individual users⁶ via cookies and does not require any registration or sign in in order to browse the tours. Therefore, we do not have any additional information about the users beyond their implicit feedback.

Based on the collected data, we evaluated two metrics: click through rate (CTR) and visit after recommend rate (VRR). CTR is a fraction between the volume of clicked and recommended objects and indicates that a recommendation was both relevant for the user and successful in catching his/her attention. VRR is a weaker criterion capturing situations, where after an object was recommended to the user, (s)he eventually visited it later on. In VRR, users might not saw recommendations, they might not fit his/her current context or the presentation was not so persuasive, however the recommended objects themselves were probably relevant). Although VRR is generally weaker than CTR, we utilized it for two primary reasons. The volume of collected feedback is considerably higher for VRR and as recommended objects were often placed outside of the initially visible area, CTR results may underestimate the true utility of recommendations. Note that if the object was recommended multiple times before the visit, we attribute the visit to the last recommendation.

4 RESULTS AND DISCUSSION

4.1 Off-line Evaluation

Our aim in off-line evaluation was threefold. First, determine whether all evaluated metrics are necessary and provide valuable additional information. Second, identify, whether there are some general trends on the sub-classes of evaluated approaches or consistently dominating recommenders and finally, select suitable candidates for on-line evaluation.

We constructed matrices of Pearson's and Spearman's correlations for all considered off-line metrics. As both matrices are highly similar, we only report Spearman's correlation (see Figure 1) to save space. The figure reveals several interesting patterns. Both diversity and rating prediction metrics are anti-correlated with ranking prediction metrics. The relation is especially strong for diversity. Novelty metrics are orthogonal to ranking accuracy as well as diversity and anti-correlated with rating prediction metrics. These

results are somewhat similar to [20], but individual clusters of metrics were less correlated in our case. We found the results of ranking vs. rating-based metrics quite consistent with findings of Herlocker et al. [9]. Metrics from rating prediction, temporal novelty, user novelty and diversity classes were highly correlated ($\rho \geq 0.96$) and therefore only one metric for each category was selected (MAE, novelty10_t, novelty10_u, ILD10). As for ranking-based metrics, results were slightly more diversified. AUC was less correlated with all other ranking accuracy metrics ($0.81 \geq \rho \geq 0.9$), while for all other metrics $\rho \geq 0.96$. Pearson's correlation further separated {nDCG100, nDCG} from the cluster of remaining ranking-based metrics and render them closer to the AUC. Therefore, we consider three clusters of ranking accuracy metrics: {AUC}, {MAP, MRR, p5, p10, r5, r10, nDCG10} and {nDCG100, nDCG}. AUC, MRR and nDCG100 metrics were selected as representatives of each cluster.⁷

We further evaluated metrics correlations for individual recommending algorithms separately. Although the results were similar in general, there were some notable differences. Novelty_u positively correlated with ranking accuracy metrics if evaluated for each recommending algorithm separately. The relation is strong especially for cosine CB recommenders. We also observed positive correlation between AUC and diversity as well as diversity and temporal novelty for cosine CB and correlation of temporal novelty with AUC for doc2vec. On the other hand, negative correlation between ranking-based metrics and both rating-based and diversity metrics was particularly strong for word2vec. Based on these observations, it may seem tempting to predict on-line performance for each algorithm separately, however the cold-start problem arises every time a new recommending algorithm has to be evaluated. Therefore, we did not follow this option and aimed on general prediction models based solely on off-line evaluation results.

Next, we evaluated individual recommender results according to the restricted set of metrics. First thing to note is that results were quite diverse. If a common ordering of metrics' results is considered (e.g., less MAE is better) 547 out of 800 recommenders were on the Pareto front. Therefore, we focused on providing some insight on recommending algorithms. Table 1 contains mean results as well as results of the best and worst member for each type of recommending algorithm. We may observe that while doc2vec models were superior in ILD, word2vec and cosine similarity performed considerably better w.r.t. ranking-based metrics. Furthermore, ILD score of doc2vec and word2vec were more than double than cosine similarity ones in average.

As for the history aggregation methods, we observed that shorter history profiles provides considerably higher user-perceived novelty score. On the other hand *max* history aggregations provided lowest novelty10_u scores in average. We also observed that slightly better results w.r.t. ranking-based metrics achieved recommenders utilizing major portion of user's history (mean, temporal, temporal-10, last-10). Furthermore, recommenders with temporal-based user profiling also exhibited higher values of novelty10_t. Both diversity and novelty enhancements considerably increased ILD and novelty10_t respectively with a negligible impact on other metrics. In general, type of the algorithm (cosine, word2vec, doc2vec) seems

⁶To be more specific, the website tracks a combination of a computer and a browser.

⁷Note that in order to illustrate differences among metrics, we occasionally display some additional ranking-based metrics in results.

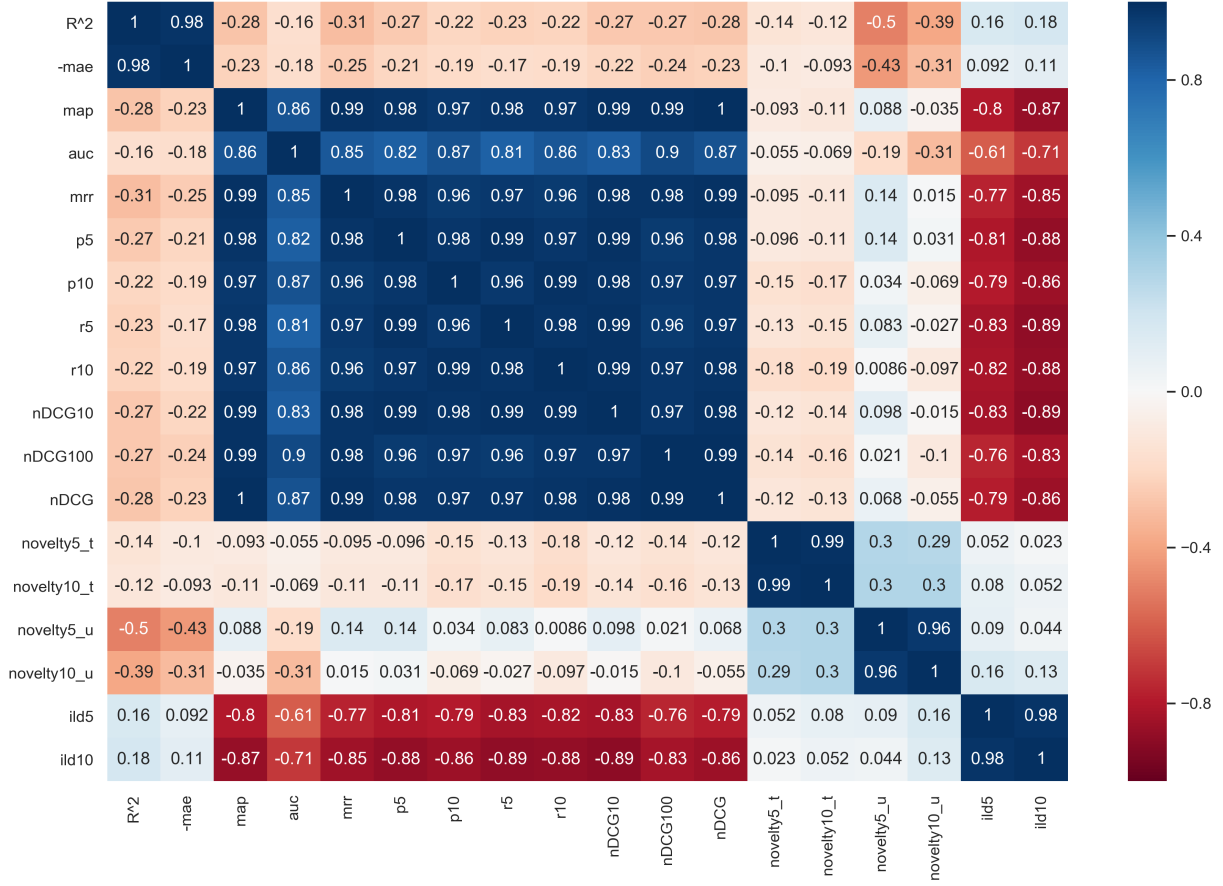


Figure 1: Spearman's correlation for off-line evaluation metrics.

to have a determining impact on ranking-based and diversity metrics, surpassing effects of history aggregation, novelty enhancements or diversity enhancements.

While selecting candidates for on-line A/B testing, our main task was to determine predictability of on-line results from off-line metrics. However, due to the limited time and available traffic, the volume of recommenders evaluated in on-line A/B testing cannot be too high.

Therefore, we adopted a following strategy: for each off-line metric, we selected the best and the worst performing recommender by default. However, if another recommender achieved close-to-best / close-to-worst performance and was already present in the set of candidates, we selected this one to save space. Furthermore, if a different type of algorithm achieved close-to-best performance, we considered its inclusion as well for the sake of diversity. Table 2 contains the final list of candidates for on-line evaluation.

4.2 On-line Evaluation

A total of 4287 users participated in the on-line evaluation, to whom, a total of 130261 objects were recommended⁸. The total volume of

⁸We excluded global-only recommendations provided to users without any past visited objects and results of users with too many visited objects (probably agency's employees).

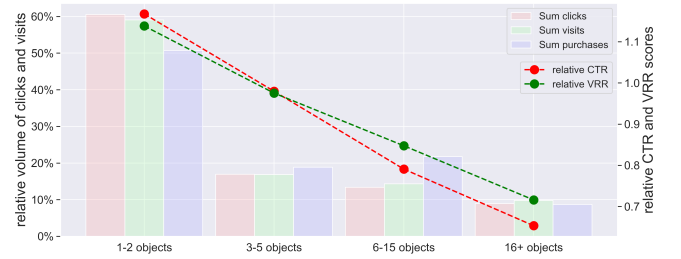


Figure 2: Comparing relative volumes of visits, clicks and purchases as well as relative CTR and VRR for different user profile sizes w.r.t. the overall values.

click-through events was 928 and the total volume of visits after recommendation was 2102. The difference between the volume of clicks and visits illustrates the problem of recommendations discoverability or ability to catch user's attention. As these features may be partially deduced from the implicit feedback data [23], we plan to incorporate off-line models that consider objects' discoverability in the future work. Another possibility is that recommended objects were potentially relevant, but not in the current context (user

Table 1: Off-line results for recommending algorithm types. Mean / Max / Min scores are depicted.

Algorithm	MAE	AUC	MRR	nDCG100	novelty10 _t	novelty10 _u	ILD10
doc2vec	0.37 / 0.46 / 0.21	0.58 / 0.72 / 0.52	0.02 / 0.06 / 0.01	0.05 / 0.10 / 0.03	0.23 / 0.30 / 0.21	0.79 / 0.91 / 0.57	0.80 / 0.89 / 0.58
cosine	0.40 / 0.42 / 0.36	0.78 / 0.80 / 0.74	0.14 / 0.19 / 0.07	0.21 / 0.24 / 0.17	0.23 / 0.27 / 0.22	0.87 / 0.97 / 0.57	0.26 / 0.44 / 0.20
word2vec	0.36 / 0.42 / 0.22	0.81 / 0.85 / 0.73	0.09 / 0.15 / 0.04	0.19 / 0.25 / 0.11	0.23 / 0.29 / 0.21	0.74 / 0.89 / 0.57	0.59 / 0.85 / 0.42

Table 2: On-line and off-line results of recommenders selected for A/B testing. Div. and Nov: stands for diversity and novelty enhancements; parameter e stands for embeddings size, w denotes context window size and s denotes whether calculating similarity on self is allowed. Best results w.r.t. each metric are in bold. For on-line metrics, results for users with 1-5 previously visited objects are depicted.

Algorithm	Parameters	History	Nov.	Div.	MAE	AUC	MRR	nDCG100	nov10 _t	nov10 _u	ild10	CTR	VRR
1: doc2vec	e: 128, w: 1	last	yes	no	0.29	0.62	0.03	0.06	0.24	0.91	0.80	0.0071	0.0171
2: doc2vec	e: 128, w: 1	temp.	no	yes	0.36	0.68	0.03	0.08	0.22	0.74	0.83	0.0079	0.0200
3: doc2vec	e: 32, w: 5	mean	no	no	0.46	0.55	0.03	0.05	0.21	0.82	0.79	0.0089	0.0179
4: doc2vec	e: 32, w: 5	mean	no	yes	0.46	0.55	0.03	0.05	0.22	0.84	0.86	0.0063	0.0151
5: doc2vec	e: 128, w: 5	max	yes	no	0.21	0.53	0.01	0.03	0.23	0.57	0.74	0.0073	0.0179
6: cosine	s:False	temp.	yes	no	0.40	0.80	0.14	0.21	0.26	0.96	0.28	0.0056	0.0092
7: cosine	s:True	mean	yes	no	0.40	0.80	0.15	0.21	0.23	0.80	0.22	0.0112	0.0218
8: cosine	s:True	last-10	no	no	0.39	0.78	0.13	0.20	0.22	0.80	0.21	0.0073	0.0166
9: word2vec	e: 64, w: 5	mean	no	yes	0.37	0.83	0.11	0.20	0.22	0.72	0.67	0.0095	0.0206
10: word2vec	e: 32, w: 5	temp.	no	yes	0.42	0.84	0.14	0.22	0.25	0.78	0.48	0.0095	0.0198
11: word2vec	e: 128, w: 3	last	no	no	0.29	0.75	0.10	0.17	0.22	0.85	0.51	0.0068	0.0173
12: word2vec	e: 32, w: 3	last-10	no	no	0.42	0.84	0.12	0.23	0.22	0.75	0.42	0.0082	0.0186

eventually process them after some time). Also this factor may be revealed by a more detailed implicit feedback analysis in the future.

While processing the results, we observed that they are strongly conditioned by the "seniority" of users measured as the volume of previously visited objects. This is illustrated on Figure 2, where four sets of users with 1 - 2, 3 - 5, 6 - 15 and 16+ previously visited objects are distinguished. The highest overall volume of interactions (clicks, visits, purchases) was collected for novice users with 1-2 visited objects. This group also exhibits highest CTR and VRR rates if compared with average values. Relative CTR and VRR drops for users with larger profiles (CTR exhibits slightly steeper decrease). On the other hand, we may see that purchase volumes did not decrease as much as other types of interactions for users with 3-15 visited objects, which shows importance of these "moderately senior" group of users from the business perspective.

Table 2 contains results of on-line A/B testing (VRR and CTR) for individual recommending algorithms and users with 1 - 5 visited objects. In general, doc2vec variants performed slightly worse than word2vec w.r.t. both CTR and VRR. Both overall best and worst algorithm belongs to the Cosine CB family. We suppose that the exceptionally bad performance of algorithm ID 6 was caused by too high user-perceived novelty (caused by $s : False$ hyperparameter). There are some related works with similar conclusions, e.g. Herlocker et al. [9] suggested that users may require a certain portion of known items to be present in the recommendations in order to trust the recommender. Also Jannach et al. [13] observed that reminders (i.e. known items) exhibit higher CTR than other forms of recommendations. Nonetheless, we plan to verify this hypothesis in the future work.

Figure 3 depicts Spearman's correlation between on-line and off-line evaluation metrics for users with 1 - 2, 3 - 5, 6 - 15 and 16+ visited objects. We may observe a significant twist in the performance according to the seniority of users. While for novel users, ranking-based metrics exhibits some correlation to both on-line metrics, this starts to decrease for more senior users (6-15 objects) and finally turns into a negative correlation for users with 16+ visited objects. An opposite behavior can be seen for user-perceived novelty and partially also for ILD.

We hypothesize that more senior users might already observed most of the straightforward choices (the evaluation site contained rather low volume of objects and provides a broad palette of browsing options). Therefore, novel and diverse suggestions may be appropriate for them. Again, we plan to focus on this hypothesis in the future work. Similarly, we plan to further investigate the rather surprising behavior of rating-based metrics as no clear pattern can be seen at the moment.

4.3 Results Post-processing

After the completion of on-line experiments, we also aimed to revisit previous off-line results with the knowledge from on-line / off-line comparison. In order to do so, we trained several simple regression methods aiming to predict CTR and VRR from off-line evaluation metrics. Because of the twist in on-line performance for increasing user profile sizes, we decided to treat the on-line results separately for for users with 1 - 2, 3 - 5 and 6 - 15 visited objects⁹. We further incorporate the user profile size into the set of

⁹We were unable to learn any reliable predictor for the group of users with 16+ objects, therefore we exclude it from the results.

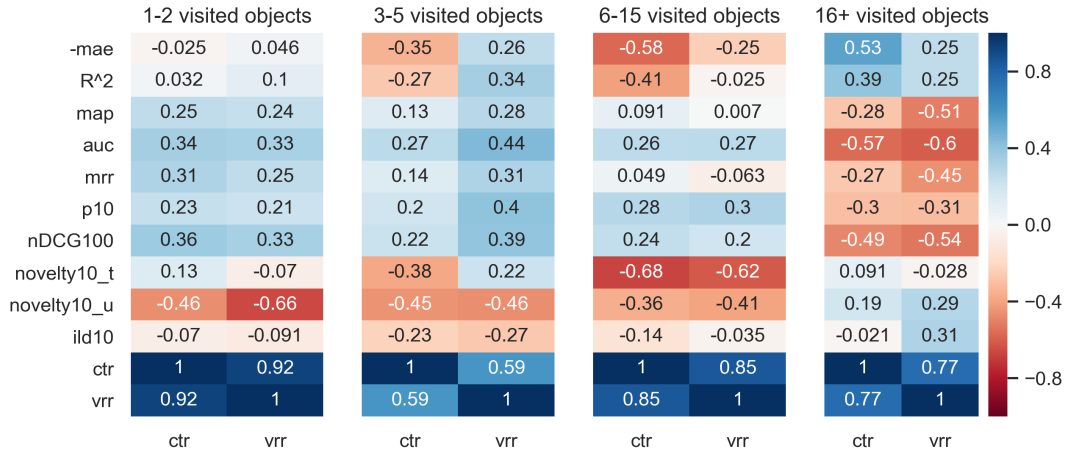


Figure 3: Spearman's correlation between off-line and on-line evaluation metrics for various user model sizes.

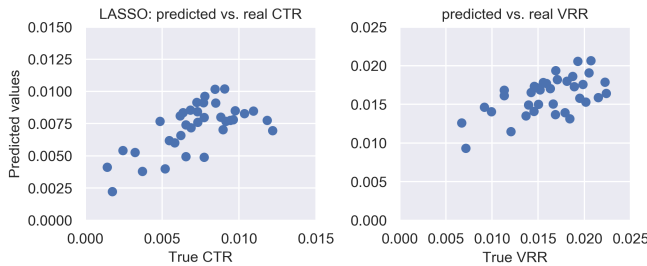


Figure 4: True values of CTR and VRR compared with the ones predicted via LASSO regression.

input variables. Due to the very small dataset size (12 algorithms \times 3 user profile size groups = 36 data points), we only focused on simple regression techniques to prevent over-fitting. We evaluated linear regression, LASSO and decision tree predictions according to the leave-one-out cross validation (LOOCV) scheme and degree-2 polynomial input feature combinations.

Due to very high coefficients, linear regression often predicted unrealistic values for both CTR and VRR (i.e. $\text{CTR} \ll 0$ and $\gg 1$). Decision tree often failed to provide reasonable predictions and (unsurprisingly) constructs large sets of algorithms with equal predicted values. However, with LASSO regression model, we were able to predict both on-line metrics up to some extent (see Figure 4. Specifically, R^2 scores were 0.42 and 0.35 for CTR and VRR respectively, while Kendal's Tau-b scores were 0.39 and 0.4 (in both cases, $p\text{-value} < 0.05$).

We also evaluated prediction of LASSO for the original set of 800 recommending algorithms. Among the top-20 results, word2vec models and cosine CB models were present, often with *max* history aggregations or some variant of *temporal* history aggregations.

Finally, in our last experiment, we aimed on verifying the quality of the CTR and VRR prediction models. Therefore, we run one more iteration of the on-line A/B testing. The best-performing model from the previous phase (ID 7 from Table 2) served as a baseline in this test. Furthermore, we included two variants (for

CTR and VRR) of the best algorithms according to LASSO regression (the individual algorithm was selected according to the actual user profile size).

Results of this experiment were unfortunately rather inconclusive. The original Cosine CB model scored 0.0064 and 0.0167 for CTR and VRR respectively. LASSO-predicted algorithms scored 0.0069 for CTR and 0.0185 for VRR. However, in both cases, the differences were not statistically significant. Nonetheless, we may conclude, that the prediction methods managed to provide candidates comparable with the so-far best method (we hope to provide more conclusive results for the camera-ready).

5 CONCLUSIONS AND FUTURE WORK

In this paper, we conducted an extensive comparison of off-line and on-line evaluation metrics in the context of small e-commerce enterprises. Experiments were held on a Czech medium-sized travel agency and shown a moderate correlation between ranking-based off-line metrics (AUC, MRR, P10, nDCG100) and both visits after recommend rate (VRR) and click-through rate (CTR) for less senior users. Similarly, results indicated a negative correlation between on-line metrics (CTR, VRR) and user-perceived novelty for the same group of users. Nonetheless, these results are reversed for the more senior users, which may indicate their saturation with simple suggestions.

However, further work is needed to verify, whether this relation may be caused by the choice of recommending algorithms, or whether there are user or object clusters with different behavior.

In addition to the direct on-line - off-line comparison, we trained several regression models aiming to predict on-line results from off-line metrics. Some of these models achieved reasonable performance for both CTR and VRR and we were able to select good additional candidate recommenders for A/B testing.

Our future work should include more detailed analysis of algorithms' off-line performance w.r.t. different segments of users and also incorporating relevant contextual information into the evaluation process. Furthermore, we plan to evaluate additional hybrid or ensemble approaches utilizing multiple sources of information (CB, CF) as well as session-based recommendations. An interesting point

to observe is, to what extent the on-line results can be predicted also for some new classes of recommending algorithms.

Our future work should also incorporate utilization of more complex implicit user feedback in order to assess importance of visited objects as well as decrease the visibility noise in on-line evaluation, especially CTR. Finally, in order to provide more transferable knowledge, we plan to perform similar experiments also on some additional small e-commerce enterprises.

ACKNOWLEDGMENTS

This paper has been supported by Czech Science Foundation (GAČR) project Nr. 19-22071Y and by Charles University project Progres Q48. Source codes, evaluation data and complete results are available from github.com/lpeska/HT2020.

REFERENCES

- [1] Fabian Abel, Yashar Deldjoo, Mehdi Elahi, and Daniel Kohlsdorf. 2017. RecSys Challenge 2017: Offline and Online Evaluation. In *RecSys '17* (Como, Italy). ACM, New York, NY, USA, 372–373. <https://doi.org/10.1145/3109859.3109954>
- [2] Ören Barkan and Noam Koenigstein. 2016. Item2Vec: Neural Item Embedding for Collaborative Filtering. *CoRR* (2016). arXiv:1603.04259
- [3] Jöran Beel and Stefan Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of *Research-Paper Recommender Systems*. In *Research and Advanced Technology for Digital Libraries*, Sarantos Kapidakis, Cezary Mazurek, and Marcin Werla (Eds.). Springer International Publishing, Cham, 153–168.
- [4] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *SIGIR '98* (Melbourne, Australia). ACM, New York, NY, USA, 335–336. <https://doi.org/10.1145/290941.291025>
- [5] Tommaso Di Noia, Iván Cantador, and Vito Claudio Ostuni. 2014. *ESWC 2014 Challenge on Book Recommendation*. Springer International Publishing, Cham, 129–143. https://doi.org/10.1007/978-3-319-12024-9_17
- [6] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Brutin, and Amr Huber. 2014. Offline and Online Evaluation of *News Recommender Systems* at Swissinfo.Ch. In *Proceedings of the 8th ACM Conference on Recommender Systems* (Foster City, Silicon Valley, California, USA) (*RecSys '14*). Association for Computing Machinery, New York, NY, USA, 169a–176. <https://doi.org/10.1145/2645710.2645745>
- [7] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *WSDM '18* (Marina Del Rey, CA, USA). ACM, New York, NY, USA, 198–206. <https://doi.org/10.1145/3159652.3159687>
- [8] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About *Playlist Recommendation Algorithms*. 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, 420a–428. <https://doi.org/10.1145/3289600.3291027>
- [9] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5a–53. <https://doi.org/10.1145/963770.963772>
- [10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- [11] Dietmar Jannach and Gediminas Adomavicius. 2016. Recommendations with a Purpose. In *RecSys '16* (Boston, Massachusetts, USA). ACM, New York, NY, USA, 7–10. <https://doi.org/10.1145/2959100.2959186>
- [12] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. 2015. Adaptation and Evaluation of Recommendations for Short-term Shopping Goals. In *RecSys '15* (Vienna, Austria). ACM, New York, NY, USA, 211–218. <https://doi.org/10.1145/2792838.2800176>
- [13] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction* 27 (2017), 351–392.
- [14] T. Joachims and F. Radlinski. 2007. Search Engines that Learn from Implicit Feedback. *Computer* 40, 8 (Aug 2007), 34–40. <https://doi.org/10.1109/MC.2007.289>
- [15] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM '17* (Cambridge, United Kingdom). ACM, New York, NY, USA, 781–789. <https://doi.org/10.1145/3018661>
- [16] Marius Kaminskas, Derek Bridge, Francelin Foping, and Donogh Roche. 2017. Product-Seeded and Basket-Seeded Recommendations for Small-Scale Retailers. *Journal on Data Semantics* 6, 1 (01 Mar 2017), 3–14. <https://doi.org/10.1007/s13740-016-0058-3>
- [17] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4 (01 Oct 2012), 441–504. <https://doi.org/10.1007/s11257-011-9118-4>
- [18] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR* (2014). <http://arxiv.org/abs/1405.4053>
- [19] Roderick J A Little and Donald B Rubin. 2002. *Statistical Analysis with Missing Data, 2nd Edition*. John Wiley & Sons, Inc., New York, NY, USA.
- [20] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems* (Vienna, Austria) (*RecSys '15*). Association for Computing Machinery, New York, NY, USA, 179a–186. <https://doi.org/10.1145/2792838.2800184>
- [21] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative Prediction and Ranking with Non-random Missing Data. In *RecSys '09* (New York, New York, USA). ACM, New York, NY, USA, 5–12. <https://doi.org/10.1145/1639714.1639717>
- [22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS '13* (Lake Tahoe, Nevada). Curran Associates Inc., USA, 3111–3119. <http://dl.acm.org/citation.cfm?id=2999792.2999959>
- [23] Ladislav Peska and Peter Vojtas. 2017. Using Implicit Preference Relations to Improve Recommender Systems. *Journal on Data Semantics* 6, 1 (01 Mar 2017), 15–30. <https://doi.org/10.1007/s13740-016-0061-8>
- [24] Pearl Pu, Li Chen, and Rong Hu. 2011. A User-centric Evaluation Framework for Recommender Systems. In *RecSys '11* (Chicago, Illinois, USA). ACM, New York, NY, USA, 157–164. <https://doi.org/10.1145/2043932.2043962>
- [25] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). 2011. *Recommender Systems Handbook*. Springer.
- [26] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results When Evaluating Recommendation Algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) (*RecSys '16*). Association for Computing Machinery, New York, NY, USA, 31a–34. <https://doi.org/10.1145/2959100.2959176>
- [27] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. Content-based Neighbor Models for Cold Start in Recommender Systems. In *RecSys Challenge '17* (Como, Italy). ACM, New York, NY, USA, Article 7, 6 pages. <https://doi.org/10.1145/3124791.3124792>