Bridging the Gap Between User-centric and Offline Evaluation of Personalized Recommendation Systems

Zainab Zolaktaf
University of British Columbia
Vancouver, Canada
zolaktaf@cs.ubc.ca

Omar AlOmeir
University of British Columbia
Vancouver, Canada
oomeir@cs.ubc.ca

Rachel Pottinger
University of British Columbia
Vancouver, Canada
rap@cs.ubc.ca

ABSTRACT

In this paper, we propose to evaluate recommender systems by conducting both offline and user-centric evaluations, while considering multiple quality aspects in realistic settings. This comprehensive evaluation would provide insight on how to improve the algorithms, and how to design better evaluation metrics, particularly for offline settings where it is cheaper to conduct evaluations. We present the preliminary offline evaluation results of several algorithms, using accuracy, novelty, and coverage metrics while considering the impact of dataset density. We propose to complement this offline evaluation with a user-centric evaluation that measures the users' perceived quality of the same algorithms.

KEYWORDS

Recommender Systems, Offline Evaluation, User-centric Evaluation

ACM Reference Format:
Zainab Zolaktaf, Omar AlOmeir, and Rachel Pottinger. 2018. Bridging the Gap Between User-centric and Offline Evaluation of Personalized Recommendation Systems. In *UMAP'18 Adjunct: 26th Conference on User Modeling, Adaptation and Personalization Adjunct, July 8–11, 2018, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3213586.3226216

1 INTRODUCTION

Recommendation systems are used in various domains ranging from movie recommendation [22], news article recommendation [15], and command or aspect recommendation for software programming [20, 21]. The goal of top-N recommendation is to to rank and suggest a set of N items to each user. Utility of the systems is commonly defined in terms of user satisfaction [1]. Accurately predicting user preference and interests, was one of the initial goals of the recommendation systems community. However, there has been a shift toward systems that consider additional aspects, like novelty or the element of surprise in their suggestions. These aspects are particularly important in exploration settings, i.e., where users need to find new items they are not aware of or could not find on their own. To determine the quality of recommendations, five recommendation aspects, and several metrics for quantifying each aspect, have been defined [2, 15]:

- **Accuracy or Error** metrics measure the relevance of suggestions by considering the user's interaction data (e.g., ratings on movies) and preferences.
- **Coverage** is the ratio of items recommended across users to the total number of items. Although coverage is a system-side objective, higher coverage can represent a more thorough investigation of the item space [7]. It can also indirectly affect user satisfaction through correlations with aspects like accuracy or novelty.
- **Diversity** measures the dissimilarity of the set of items recommended to the user.
- **Novelty** measures whether the item is new for the user. As explained in [11] an item can be novel in three ways: it is new to the system and is likely to be unseen by most users (cold-start) 2) it existed in the system, but is new to a single user 3) it existed in the system, was previously known to the user, but is a forgotten item.
- **Serendipity** considers both the novelty and accuracy (or usefulness) of recommendations [15].

Typically, a new algorithm's performance is measured in an offline setting, using a subset of these aspects, e.g., accuracy, novelty, and coverage [22], on existing datasets like MovieLens [8] or Netflix from the movie recommendation domain. However, because each recommendation aspect can be measured using several metrics, there is a choice of which metrics to use for measuring an aspect. For example, recommendation accuracy can be quantified using error metrics like Mean Absolute Error, Root Mean Square Error, or accuracy metrics like F-measure and recall [2]. Although offline studies are cheap and easy to conduct, perhaps their most significant drawback is that it is unclear if and how much the offline metric measurements conform with users' perception of the same aspects.

An alternative, albeit more expensive option, is to perform user-centric evaluations and study recommendation performance through interactions with real users [2, 12]. In [2], user-centric evaluations are broadly categorized into user studies and online evaluations where a few of the best recommendation systems are deployed. Prior works [4, 16] that use a combination of offline and user-centric evaluation have confirmed the positive impact of aspects like accuracy, serendipity, and novelty, on user's perceived quality of recommendations. However, these studies have focused on comparing offline accuracy measurements against user-centric notions of quality, e.g., novelty and accuracy. The practicality of other offline metrics is less studied [4]. E.g., it is unclear whether offline novelty measurements comply with user-centric novelty measurements.

Recent work has also examined whether online user engagement metrics can be predicted from offline performance metrics [15]. In [15], click-through rates (CTR) and online accuracy were used as

UMAP'18 Adjunct, July 8–11, 2018, Singapore, Singapore

ACM ISBN 978-1-4503-5784-5/18/07…$15.00
https://doi.org/10.1145/3213586.3226216

indicators of user engagement and online performance. They concluded that the significance of offline metrics in predicting online performance depends on the algorithms, properties of the datasets, and the online performance metric. However, this study only considers the news article domain, and studies three algorithms.

In summary, there is a gap between offline and user-centric evaluations of recommender systems, in terms of both the metrics that are used, and the settings under which evaluations are conducted. Given an appropriate set of algorithms, and fixing the domain of interest, e.g., movie domain where users provide ratings on movies, there are several open research questions that remain to be addressed. Answering these questions could potentially help bridge the gap between user-centric and offline evaluation:

- RQ1. For each of the recommendation aspects, several offline evaluation metrics have been defined [2]. How do these metrics correlate with the user perceived quality for the same aspect, and which metrics are most appropriate?
- RQ2. Dataset *density* is the ratio of available observations to the total possible observations. How does density affect recommendation performance, the aspect measurements and their correlations, in offline and user-centric evaluations?
- RQ3. What are the best online performance or user engagement indicators?

We propose to address these questions in a joint offline and user-centric evaluation. This study would help establish the importance of the offline criteria from the perspective of users, and provide insights regarding the differences between offline and user-centric measurements. This paper is organized as follows: in Section 2, we describe a preliminary study of the offline components of RQ1 and RQ3, examining the performance of several algorithms while considering the impact of dataset density. In Section 3, we propose to complement the offline results with online evaluations and user studies that address RQ1, and RQ2, and RQ3. We conclude in Section 4.

2 OFFLINE EVALUATION

We focus on the offline components of RQ1 and RQ3 in this section. We study the performance of several algorithms while considering the impact of dataset density.

2.1 Compared Algorithms

We compare the following algorithms:

- **Rand** is non-personalized and randomly suggests N unseen items from among all items. It obtains high coverage and novelty, but low accuracy [18].
- **Pop** [5] is a non-personalized and suggests the most popular unseen items. For ranking tasks, it obtains high accuracy [5, 18], since it takes advantage of the popularity bias of the data. However, Pop makes trivial recommendations that lack novelty [5].
- **Regularized SVD** (RSVD) [13] is a latent-factor model for rating prediction. We used `LIBMF`, with L2-Norm as the loss function, and L2-regularization, and Stochastic Gradient Descent for optimization. We performed 10-fold cross validation and tested: number of latent factors $g \in \{8, 20, 40, 50, 80, 100\}$, L2-regularization coefficients $\lambda \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, learning rate $\eta \in \{0.002, 0.003, 0.01, 0.03\}$. For each dataset, we used the parameters that led to best performance w.r.t. loss function (details in [22]).
- **Pure SVD** [5] is a latent factor model, known for achieving high accuracy and novelty [5] in ranking tasks. In Pure SVD, missing values are imputed by zeros and conventional SVD is performed. We used Python's `sparsesvd` module and tested two configurations: one with 10 latent factors (PSVD10), and one with 100 latent factors (PSVD100).
- **CoFiRank** [19] is a ranking prediction model that can optimize directly the Normalized Discounted Cumulative Gain ranking measure and regression loss. We found the latter performed better in our experiments on ML-100K and ML-1M. We report results for a model with 10 latent factors (CofiR10) and one with 100 latent factors (CofiR100). We used the source code from [19].
- **GANC** [22] is a generic re-ranking framework for trading-off accuracy, novelty, and coverage. The main idea is to estimate for each user a long-tail novelty preference value directly from interaction data. We use PSVD100 as the base accuracy recommender, with dynamic coverage, $|S| = 900$, and generalized preference.

Dataset	$\|\mathcal{D}\|$	$\|\mathcal{U}\|$	$\|\mathcal{I}\|$	d%	$\mathcal{L}$%	κ	τ
ML-100K	100K	943	1682	6.30	66.98	0.5	20
ML-1M	1M	6,040	3,706	4.47	67.58	0.5	20
ML-10M	10M	69,878	10,677	1.34	84.31	0.5	20
MT-200k	172,506	7,969	13,864	0.16	86.84	0.8	5
Netflix	98,754,394	459,497	17,770	1.21	88.27	-	-

Table 1: Datasets description. $|\mathcal{D}|$ is number of ratings in dataset. Density is d% $= |\mathcal{D}|/(|\mathcal{U}| * |\mathcal{I}|) \times 100$%. Long-tail percentage is $\mathcal{L}\% = (|\mathcal{L}|/|\mathcal{I}^{\mathcal{R}}|) \times 100$%. Train-test split ratio per user is κ, τ is the minimum number of ratings per user.

2.2 Experimental Setup

2.2.1 Datasets and data split. Table 1 describes our datasets. We use MovieLens 100K (ML-100K), 1 Million (Ml-1M), 10 Million (ML-10M) [8], Netflix, and MovieTweetings 200K (MT-200K) [6]. In the MovieLens datasets, every consumer has rated at least 20 movies ($\tau = 20$), with $r_{ui} \in \{1, \ldots, 5\}$ (ML-10M has half-star increments). MT-200K contains voluntary movie ratings posted on twitter, with $r_{ui} \in \{0, \ldots, 10\}$. Following [9], we preprocessed this dataset to map the ratings to the interval $[1, 5]$. Due to the extreme sparsity of this dataset and to ensure every user has some data to learn from, we filtered the users to keep those with at least 5 ratings ($\tau = 5$).

Our selected datasets have varying density levels. Additionally, MT-200K and Netflix include a large number of difficult infrequent users, i.e., in MT-200K, 47.42% (3.37% in Netflix) of the users have rated fewer than 10 items, with the minimum being 4 ratings. Next, we randomly split each dataset into train and test sets by keeping a fixed ratio κ of each user's ratings in the train set and moving the rest to the test set [14]. For ML-1M and ML-10M, we set $\kappa = 0.5$. For MT-200K, we set $\kappa = 0.8$. For Netflix, we use their probe set as our test set, and remove the corresponding ratings from train (details in [22]).

Local Ranking Accuracy Metrics	Precision@N = $\frac{1}{N\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u^{\mathcal{T}+}\cap\mathcal{P}_u\rvert$
	Recall@N = $\frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\frac{\lvert\mathcal{I}_u^{\mathcal{T}+}\cap\mathcal{P}_u\rvert}{\lvert\mathcal{I}_u^{\mathcal{T}+}\rvert}$
	F-measure@N = $\frac{\text{Precision@N.Recall@N}}{\text{Precision@N+Recall@N}}$
Longtail Promotion	LTAccuracy@N = $\frac{1}{N\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\lvert\mathcal{L}\cap\mathcal{P}_u\rvert$
	StratRecall@N = $\frac{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u^{\mathcal{T}+}\cap\mathcal{P}_u}\left(\frac{1}{f_i^{\mathcal{R}}}\right)^\beta}{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u^{\mathcal{T}+}}\left(\frac{1}{f_i^{\mathcal{R}}}\right)^\beta}$
Coverage Metrics	Coverage@N = $\frac{\lvert\cup_{u\in\mathcal{U}}\mathcal{P}_u\rvert}{\lvert\mathcal{I}\rvert}$
	Gini@N = $\frac{1}{\lvert\mathcal{I}\rvert}\left(\lvert\mathcal{I}\rvert+1-2\frac{\sum_{j=1}^{\lvert\mathcal{I}\rvert}(\lvert\mathcal{I}\rvert+1-j)f[j]}{\sum_{j=1}^{\lvert\mathcal{I}\rvert}f[j]}\right)$

Table 2: Performance Metrics. Notation is in Section 2.2.2.

2.2.2 Test ranking protocol and performance metrics. We adopt the "All unrated items test ranking protocol" [17, 18] where for each user u, we generate the top-N set $\mathcal{P}_u$, by ranking all items that do not appear in the train set of that user.

In our offline evaluation, we focused on three aspects: accuracy, novelty, and coverage. Table 2 summarizes the performance metrics. Let $\mathcal{U}$ denote the set of users, $\mathcal{I}$ the set of items, $\mathcal{R}$ the train set, and $\mathcal{T}$ the test set. To define the long-tail items, $\mathcal{L}$, we sort the items in decreasing order of their observed train set ratings. Using the Pareto principle, or the 80/20 rule, we define $\mathcal{L}$ as those that generate the lower 20% of the total ratings in the train set [22].

For accuracy, we use local rank-based precision and recall [1, 17, 18], where each metric is computed per user and then averaged across all users. Precision is the proportion of relevant test items in the top-N set, and recall is the proportion of relevant test items retrieved from among a user's relevant test items. For each user u, the relevant test items are those that she rated highly, i.e., $\mathcal{I}_u^{\mathcal{T}+} = \{i : i \in \mathcal{I}_u^{\mathcal{T}}, r_{ui} \geq 4\}$ [1], f. Note, because the collected datasets have many missing ratings, the hypothesis that only the observed test ratings are relevant, underestimates the true precision and recall [17]. But, this holds for all algorithms, and the measurements are known to reflect performance in real-world settings [17]. F-measure is the harmonic mean of precision and recall.

For novelty, we use Long-Tail Accuracy (LTAccuracy@N) [10] which is the proportion of the recommended items that are unlikely to be seen by the user. Moreover, we use Stratified Recall (StratRecall@N) [17] which measures the ability of a model to compensate for the popularity bias of items w.r.t train set. Similar to [17], we set $\beta = 0.5$. Note, LTAccuracy emphasizes both novelty and coverage, while Stratified Recall emphasizes novelty and accuracy.

For coverage we use two metrics: Coverage@N is the ratio of the total number of distinct recommended items to the total number of items [10, 18]. A maximum value of 1 indicates each item in $\mathcal{I}$ has been recommended at least once. Gini [10, 22] measures the inequality among values of a frequency distribution $\mathbf{f}$. It lies in $[0, 1]$, with 0 representing perfect equality, and larger values representing skewed distributions. In Table 2, $\mathbf{f}$ is the recommendation frequency of items, and is sorted in non-decreasing order, i.e., $f[j] \leq f[j+1]$.

2.3 Results

In each of the sub-figures in Figure 1, the datasets are ordered in increasing density on the x-axis. For F-measure vs density, the ranking of algorithms in the most dense setting (ML-100k with density of 6.30) differs from their ranking in sparse settings (MT-200K with density 0.16). However the ranking of the algorithms w.r.t. coverage and gini is relatively stable. Most algorithms target accuracy and under-perform w.r.t. other metrics. GANC targets trade-offs between coverage, novelty, and accuracy. To better understand which offline metrics are most useful, and which algorithms are preferable by users, particularly in sparse settings where users provide little feedback, user-centric evaluations are needed.

3 USER-CENTRIC EVALUATION

Next, we propose user-centric solutions for the research questions posed in Section 1. We propose conducting a user study that includes a questionnaire, to interpret and quantify the correlation between offline and user-centric aspects measurements (RQ1). Our objective is to directly ask easy to understand questions that are closely related to the offline metrics. For example, the following questions can be asked: (1) Accuracy: How many of the recommended items reflect your interests? (2) Novelty: How many of the recommended items are you aware of? Among the items you are unaware of, how many of them can you find on your own? (3) Serendipity: Among the items you were unaware of, how many are you interested in watching? (4) Overall satisfaction: Which set of recommended items did you like best? Provide a satisfaction score for each set.

For guidance, we will rely on previously developed frameworks that target user-centric evaluations of recommender systems [12, 16]. Qualitative answers from the user study can help us find a meaningful relation between the offline aspect measurements and quantitative results of the questionnaire. As future work, however, we need to devise an adequate mechanism for comparing (and quantifying) the outcome of a potential questionnaire with the offline study.

In order to examine the effect of dataset density on offline and user-centric evaluations (RQ2), one option is to conduct a controlled lab experiment. In particular, we can use either the methodology in Section 2, where the datasets have varying density, or we can sub-sample a single dataset to create multiple datasets with varying density. The algorithms in Section 2, can then be trained on each dataset, and deployed for making recommendations. In this experiment, we need to evaluate the recommendations offered by 6 different algorithms at 5 different levels of density. This leads to a 6×5 factorial design.

Furthermore, as shown in [3], asking users to rate items has a minimal effect on overall satisfaction. It also shows a minimal effect of profile length on users' perception of quality. Thus, collecting ratings from users and adding profile length as an extra variable can be an optional step. However, it can lead to a more complex experiment design and a longer more fatigue-inducing study.

Regarding the format of the lab experiment, we can conduct either an in-lab or crowd-sourced experiment. While the former allows us to interview and examine the preferences of the participants in detail, the latter allows us to recruit more participants.

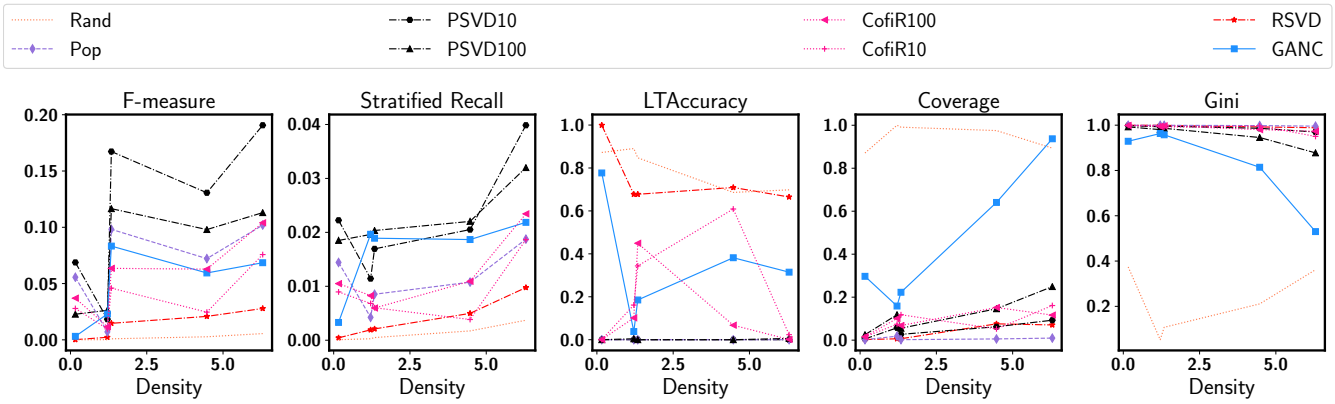

Figure 1: Offline evaluation results from accuracy, novelty, and coverage aspects, as density of datasets varies.

For the in-lab experiment, it can be difficult and costly to achieve reliable and statistically significant results. Thus, we may need to consider fewer levels for each independent variable. On the other hand, the results of the the crowd-sourced lab experiment may be statistically significant, but less informative.

A more appropriate type of evaluation for RQ2 may be to analyze usage logs. Here, a few of the best recommendation systems are deployed online and user interactions are recorded. This logged data may include CTR, online accuracy, system usage time, views, clicks, likes, shares, and comments on recommendations [1]. Given the usage logs, we would need to conduct a thorough analysis to interpret and quantify the relation between user-centric and offline aspect measurement, and to measure the impact of the recommendations.

In order to do this, we must also identify key performance indicators (RQ3). Question 4 in our user study, helps identify a single metric to measure users' overall satisfaction with the system (RQ3), particularly for top-N recommendation scenarios. Answers to these types of subjective questions can be compared and contrasted with logged interaction data, to help identify some of the key engagement indicators.

4 CONCLUSION

In this paper, we proposed to measure the quality of recommender systems using closely related offline and user-centric evaluations in realistic settings. This approach includes conducting offline evaluations, user studies, and online evaluations using similar aspects. We propose using multiple metrics for each of the recommendation aspects while considering considering the impact of dataset density. We also suggest a number of ideas for running online evaluations and user studies. The next step would be to design the experiments taking into account all the different variables and possible hypotheses. We hope our proposed ideas can work as building blocks that lead to a better understanding of how to evaluate the performance of recommender systems with end-users in mind.

REFERENCES

[1] Deepak K. Agarwal and Bee-Chung Chen. 2016. *Statistical Methods for Recommender Systems.* Cambridge University Press.

[2] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. *Recommender Systems Handbook.* Springer US, Chapter Novelty and Diversity in Recommender Systems.

[3] Paolo Cremonesi, Francesco Epifania, and Franca Garzotto. 2012. User profiling vs. accuracy in recommender system user experience. *AVI* (2012).

[4] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Papadopoulos, and Roberto Turrin. 2011. Comparative evaluation of recommender system quality. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems.* ACM.

[5] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys.*

[6] Simon Dooms, Toon De Pessemier, and Luc Martens. 2013. Movietweetings: a movie rating dataset collected from twitter. In *CrowdRec at RecSys.*

[7] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *RecSys.*

[8] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *ACM Trans. on Interactive Intelligent Systems (TiiS)* (2016).

[9] Jose M Hernandez-lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *ICML.*

[10] Yu-Chieh Ho, Yi-Ting Chiang, and Jane Yung-Jen Hsu. 2014. Who likes it more?: mining worth-recommending items from long tails by modeling relative preference. In *WSDM.*

[11] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A Konstan, and Paul Schrater. 2015. I like to explore sometimes: Adapting to dynamic user novelty preferences. In *RecSys.*

[12] 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* (2012).

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[14] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2014. Local collaborative ranking. In *WWW.*

[15] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *RecSys.*

[16] Pearl Pu, Li Chen, and Rong Hu. 2011. A user-centric evaluation framework for recommender systems. *Recsys* (2011).

[17] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *RecSys.*

[18] Saúl Vargas and Pablo Castells. 2014. Improving sales diversity by recommending users to items. In *RecSys.*

[19] Markus Weimer, Alexandros Karatzoglou, Quoc V Le, and Alex J Smola. 2008. Cofi rank-maximum margin matrix factorization for collaborative ranking. In *NIPS.*

[20] Sedigheh Zolaktaf and Gail C Murphy. 2015. What to learn next: Recommending commands in a feature-rich environment. In *ICMLA.*

[21] Zainab Zolaktaf. 2017. Facilitating User Interaction With Data. In *PhD@VLDB.*

[22] Zainab Zolaktaf, Reza Babanezhad, and Rachel Pottinger. 2018. A Generic Top-N Recommendation Framework For Trading-off Accuracy, Novelty, and Coverage. In *ICDE.*