

Evaluating Music Recommendations with Binary Feedback for Multiple Stakeholders

SASHA STOIKOV, Cornell Financial Engineering, Cornell Tech, USA

HONGYI WEN, Cornell Information Science, Cornell Tech, USA

High quality user feedback data is essential to training and evaluating a successful music recommendation system, particularly one that has to balance the needs of multiple stakeholders. Most existing music datasets suffer from noisy feedback and self-selection biases inherent in the data collected by music platforms. Using the Piki Music dataset of 500k ratings collected over a two-year time period, we evaluate the performance of classic recommendation algorithms on three important stakeholders: *consumers, well-known artists and lesser-known artists*. We show that a matrix factorization algorithm trained on both likes and dislikes performs significantly better compared to one trained only on likes for all three stakeholders.

Additional Key Words and Phrases: Datasets, Music recommendations, Multi-stakeholders

1 INTRODUCTION

Music recommendation algorithms play a major role in what music gets listened to on streaming platforms [15, 16]. This in turn influences which artists make a living from streaming and which ones do not. Understanding the mechanisms that cause an algorithm to push certain artists ahead of others is increasingly urgent. In this paper, we focus on three stakeholders of a music recommendation system: *music consumers, well-known artists and lesser-known artists*.

Much has been written about competing classes of algorithms and metrics. Unfortunately, the growing number of proposed metrics makes it difficult to answer straightforward questions related to the satisfaction of the stakeholders of a music streaming platform. For example, what proportion of recommendations is actually liked by users? Do the algorithms serve well-known artists better than lesser-known artists? Are music consumers more or less satisfied with well-known or lesser-known recommendations? There is often a tension between metrics that aim to measure familiarity, relevance and predictability, and metrics that aim to measure fairness, diversity and serendipity. Combining these metrics into objective functions that represent the interests of the aforementioned stakeholders is challenging.

At the heart of this problem are three common assumptions in the data used for training and testing recommendation algorithms: (i) unheard songs are assumed to be disliked, though in reality unheard songs are often excellent and (ii) played songs are assumed to be liked, though they may have been passively listened to on a playlist (iii) rated songs are randomly presented to users, while they are often self-selected by the user. All of these assumptions tend to favor the well-known artists, who are more often heard, more often recommended on playlists and more often remembered at the search bar. To mitigate these concerns we built Piki ¹, a music discovery tool that collects ratings through a binary choice (like/dislike) on song clips that cannot be searched or skipped. We present the *Piki Music* dataset with the goal of enabling researchers and practitioners from the RecSys community to explore the above challenges. We highlight the following contributions:

^{*}Copyright 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
Presented at the MORS workshop held in conjunction with the 15th ACM Conference on Recommender Systems (RecSys), 2021, in Amsterdam, Netherlands.
¹www.piki.nyc

Authors' addresses: Sasha Stoikov, Cornell Financial Engineering, Cornell Tech, New York, USA, sfs33@cornell.edu; Hongyi Wen, Cornell Information Science, Cornell Tech, New York, USA, hw557@cornell.edu.

- We collect the Piki Music dataset in a way that incentivizes users to give truthful binary ratings to songs, thus mitigating the self-selection and noisy feedback biases inherent in most recommendation datasets. (Section 3).
- We quantify the value of a dislike by training a matrix factorization algorithm on this binary dataset. We find that it performs significantly better than when trained on positive-only feedback for all three stakeholders (Section 4).

2 RELATED WORK

Datasets with Explicit and Implicit User feedback. Datasets consisting of user-item interactions are typically elicited in two ways: explicitly or implicitly. Explicit feedback refers to an action that a user performs with the intention of giving their opinion on the quality of an item, for example, giving a rating to a watched movie. Two well-known datasets for recommendations, the Netflix Prize dataset [3] and the Movielens dataset [8], are examples of explicit feedback datasets. Both consist of 1-5 star ratings for millions of user-item pairs. Explicit feedback is voluntary, so this kind of dataset is often subject to self-selection bias: if there is no obligation to rate, users tend to rate only when they feel very strongly about an item, and rate more items they like than items they dislike. Unbalanced explicit training datasets have been documented in the literature and some flattening techniques [14] have been shown to improve recommendation quality.

Implicit feedback refers to an action that a user performs with an intention other than giving their opinion on the quality of the item, such as clicks, purchases, etc. Implicit feedback collected by streaming apps is commonly used to train modern recommendation systems [10]. However, researchers have expressed concerns about training and evaluating algorithms using this type of feedback data [13, 20]. A major problem with implicit feedback datasets for music recommendations, such as those obtained on Lastfm [4] or Spotify [5], is that they may measure a noisy signal of the user’s true preferences. If a user listens to a song, that action may be interpreted as positive feedback, though the song may have been passively played as background music [20]. Moreover, implicit feedback typically consists of “positive-only” data. To train and test an algorithm, missing user-item interactions are typically labeled as negative feedback, which only adds more noise to the training data. One approach to deal with implicit data is to label songs that are less often played as negative feedback [19]. However, few datasets with explicit positive and negative user preferences are publicly available, making it hard to investigate the limitations of implicit feedback.

Recommendations for multi-stakeholders. A recent paper calls for a paradigm shift from user-centric metrics towards modeling the value that recommendation systems bring to their multiple stakeholders [11]. They identify potential stakeholders as consumers, producers, platforms and society at large and call for evaluation designs aligned with the goals of each stakeholder. The way forward, they suggest, is to design better evaluation methods that combine multiple goals and generalize beyond domain specific applications. Researchers have investigated recommendations from the multi-stakeholder perspective by optimizing over multiple objectives such as diversity, relevance, fairness, satisfaction [15, 18]. They highlight an inherent tension between relevant recommendations that are similar to past consumption and diverse recommendations that are outside of a user’s echo chamber. Fairness in compensating for the popularity bias has gathered particular attention [1, 6, 12]. The authors suggest addressing multiple stakeholders by modeling profit-aware recommendations [2, 9], e.g., recommendations that are linked to sales. However, a challenge is the lack of publicly-available data with multi-stakeholder characteristics, which tends to be sensitive data.

In the music domain, lesser-known artist have expressed many concerns, which include reaching an audience, transparency in recommendations, localizing discovery, gender balance and popularity bias, according to a qualitative study [7]. In the sequel, we take a quantitative approach to some of these concerns with the Piki Music Dataset.

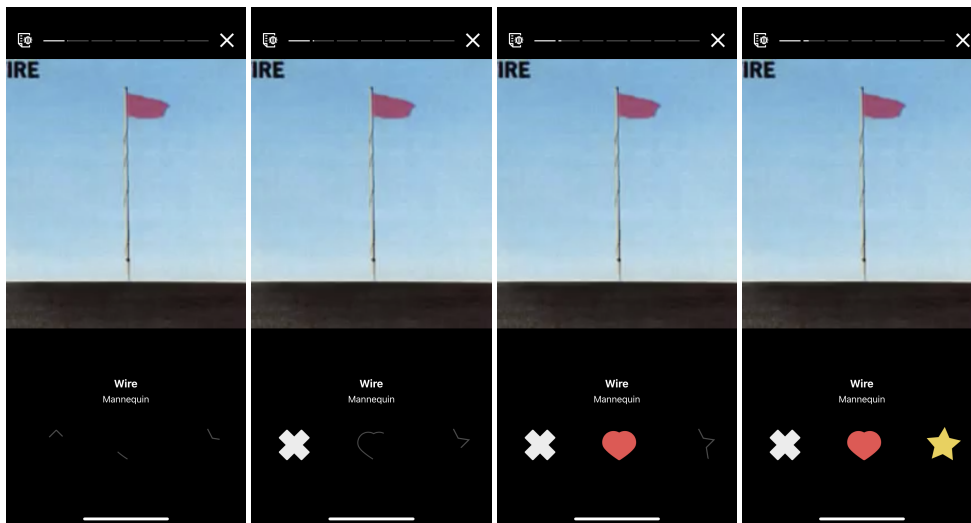


Fig. 1. The Piki Music App interface. The dislike button is unlocked after 3 seconds, the like button after 6 seconds and the superlike button after 12 seconds. This incentivizes the users to vote truthfully: to dislike is easy but in order to like, a user must invest time in the song.

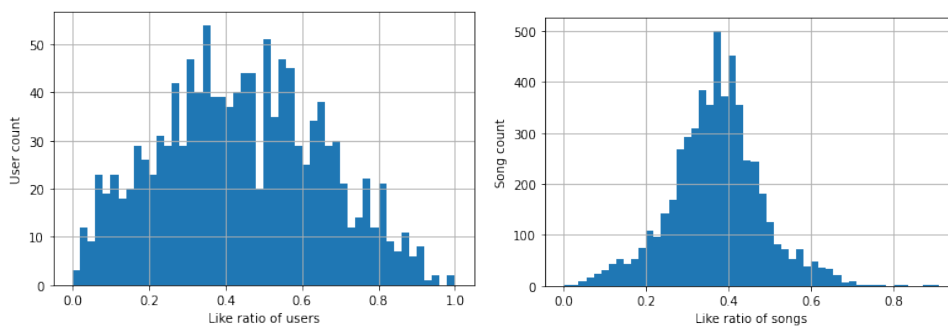


Fig. 2. The distribution of like ratios across users and songs.

3 PIKI MUSIC DATASET

Through the Piki interface (Fig. 1), we collect binary data while incentivizing users to provide feedback in a way that is aligned with their individual tastes. The Piki Music dataset currently consists of 2723 anonymized users, 66,532 anonymized songs and 500K binary ratings and the data collection is on-going. Figure 2 illustrates the distribution of like rates across users and songs. The columns of the dataset are as following:

- timestamp: a datetime variable
- user_id: an anonymized user id
- song_id: an anonymized song id

- liked: this is the binary indicator, 1 if the song is liked, or 0 if the song is disliked. Note that the feedback consists of 39% likes and 61% dislikes. The superlike indicator, labeled 2, is included in the data, though we treat it as a like in our experiments.
- personalized: this is 1 if the song was recommended based on their previous choices or 0 if the song was selected randomly. Note that the songs recommended are 66% personalized and 34% random songs. We have included this flag in the dataset, to allow mitigation of the recommendation bias of the data, though this question is not within the scope of our study.
- spotify_popularity: this is the song’s artist’s popularity, a value between 0 and 100, with 100 being the most popular. It is published by Spotify for each artist, through their publicly-available API ². The average value of the Spotify popularity in our data set is 52, so we classify songs as coming from well-known artists if the value is above this mean and as a lesser-known artist if it is below the mean. Note that this threshold corresponds to artists that have approximately 350,000 monthly listeners, which on average generates around \$2000 per month, assuming this is a solo artist without a label.

3.1 Binary data collection

Users on Piki provide explicit binary feedback, by liking or disliking music clips. Users do not have access to a search bar and thus cannot control what songs they will hear. They are asked to like or dislike 30 second music video clips. Figure 1 shows how the interface presents the songs in batches, much like a social media story. The binary nature of the Piki music data set addresses our first concern with the training data, namely, that we won’t need to treat unheard songs as disliked, since we have a set of disliked songs to train and test the algorithm.

3.2 Incentives to vote truthfully

The dislike button is unlocked after 3 seconds (see the first and second images in Fig. 1), the like button is unlocked after 6 seconds (see the 3rd image in Fig. 1) and the superlike button is unlocked after 12 seconds (see the 4th image in Fig. 1). The clip starts 40 seconds into the song and when the clip is over, 30 seconds later, the user may replay the clip before rating it. This mechanism aligns the users’ ratings with their preferences. The 3 second lock period for the dislike button ensures that all songs are given a fair chance. The 6 and 12 second lock periods guarantee that positive ratings are backed up by a meaningful time investment in the song. Only songs that truly capture a user’s attention get liked or superliked. Piki users are rewarded with micropayments each time they complete a set of ratings. Both the timing and the financial rewards help mitigate our second data concern, namely that liked songs are actively liked, not just played passively on a playlist.

4 EXPERIMENTS

We split the dataset into a training set \mathcal{T} and an evaluation set \mathcal{E} using random sampling according to 80%/20% splits stratified by user and average the results across 5 runs. Algorithms are trained on \mathcal{T} and scores are computed on interactions in \mathcal{E} .

²<https://developer.spotify.com/documentation/web-api/reference/category-artists>

4.1 Training matrix factorization algorithms

Given a set of users \mathcal{U} , songs \mathcal{I} and ratings \mathcal{R} , collaborative filtering aims to learn d -dimension latent user vectors $x \in \mathbb{R}^{|\mathcal{U}| \times d}$ and latent item vectors $y \in \mathbb{R}^{|\mathcal{I}| \times d}$ from the sparse user-item rating matrix through singular value decomposition (SVD) [17]. Predicted user preference scores are given by the dot products between user and item vectors:

$$\hat{p}_{ui} = x_u^T y_i. \quad (1)$$

A classic algorithm from this framework is the Weighted Regularized Matrix Factorization (WRMF) [10]. Specifically, WRMF optimizes for the following objective:

$$\min_{x^* y^*} \sum_{u,i} c_{ui} (r_{ui} - \hat{p}_{ui})^2 + \lambda R(x, y), \quad (2)$$

where r_{ui} is the ground true preference score for user u to item i , c_{ui} is the weight put on each observation, $R(x, y) = \|x\|_f^2 + \|y\|_f^2$ represents the Frobenius norms for regularizing user and items matrices, and λ is regularization parameter. Note that with implicit feedback datasets, r_{ui} is assumed to be $\{0, 1\}$, where a click is treated as 1 and missing data from the rating matrix is treated as 0.

In a setting where negative feedback is available, a generalized framework is proposed to incorporate negative feedback during training [20]. The objective function can be written as:

$$\min_{x^* y^*} \alpha \sum_{(u,i) \in \mathcal{T}_P} (1 - \hat{p}_{ui})^2 + \beta \sum_{(u,i) \in \mathcal{T}_N} \hat{p}_{ui}^2 + \gamma \sum_{(u,i) \in \mathcal{T}_M} \hat{p}_{ui}^2 + \lambda R(x, y), \quad (3)$$

where $\mathcal{T}_P, \mathcal{T}_N, \mathcal{T}_M$ refer to positive, negative and missing user feedback, α, β, γ are weights assigned to the corresponding sets of user feedback. We highlight two types of weight schemas:

- **WRMF with Likes:** $\alpha = \gamma = 0.5, \beta = 0$. This means that we only sample from the positive and missing feedback, while ignoring the negative feedback.
- **WRMF with Likes and Dislikes:** $\alpha = \beta = 0.5, \gamma = 0$. This means that we only sample from positive and negative feedback, without the need to sample from missing data.

4.2 Implementations

We implemented the WRMF algorithm based on the OpenRec library [21]. We used the Adagrad optimizer with a learning rate of 0.01 and a batch size of 512 to train the model. The regularization parameter λ is tuned on a validation set from $\{0.1, 0.01, 0.001, 0.0001\}$ and early stopping is performed. We used a dimension of $d = 20$ for both user and item latent factors. The code for experiments and the Piki Music Dataset is public ³.

4.3 Performance metrics

The evaluation dataset \mathcal{E} can be segmented into $\mathcal{E}_P, \mathcal{E}_N, \mathcal{E}_M$ referring to positive, negative and missing user feedback. Popular metrics like Recall@ k and Precision@ k aim to quantify the relevance of playlists of length k . However, they implicitly assume that \mathcal{E}_N and \mathcal{E}_M are indistinguishable from each other.

For a trained algorithm, all scores above a given threshold are classified as recommendations. Without loss of generality, we use the median of scores from model outputs as the threshold in our experiments. We use \mathcal{R} to denote the songs recommended by an algorithm from \mathcal{E} . The set of recommended songs from the well-known artists is \mathcal{R}_w

³<https://github.com/sstokov/piki-music-dataset>

Model	Well-known artists(%)	Lesser known artists (%)	Consumers (%)
Popularity (recommend well-known)	41.1±0.04	-	41.1±0.04
Anti-popularity (recommend lesser-known)	-	36.3±0.06	36.3±0.06
WRMF with Likes	51.9±0.29	48.2±0.35	50.1±0.28
WRMF with Likes and Dislikes	61.3±0.18	57.6±0.46	59.6±0.30

Table 1. Performance measured by precision for different stakeholders under WRMF models.

and the set of recommended songs from the lesser-known artists is \mathcal{R}_l . It is obvious that $\mathcal{R} = \mathcal{R}_w \cup \mathcal{R}_l$. For each song in \mathcal{R} , we have a corresponding binary rating of $\{0, 1\}$ from the consumers. For simplicity, we use $S_{\mathcal{R}}$ to represent the vector for binary ratings on songs in \mathcal{R} . Similarly, we have $S_{\mathcal{R}_w}$ and $S_{\mathcal{R}_l}$.

- **Consumers:** the proportion of the recommended songs from the evaluation set that are actually liked by users: $\frac{\sum S_{\mathcal{R}}}{|\mathcal{S}_{\mathcal{R}}|}$. The intuition is that a higher precision is aligned with better user experience for the consumers.
- **Well-known artists:** the proportion of recommendations coming from the well-known artists that are actually liked by users: $\frac{\sum S_{\mathcal{R}_w}}{|\mathcal{S}_{\mathcal{R}_w}|}$. A higher precision leads to more effective song exposure for well-known artists.
- **Lesser-known artists:** the proportion of recommendations coming from the lesser-known artists that are actually liked by users: $\frac{\sum S_{\mathcal{R}_l}}{|\mathcal{S}_{\mathcal{R}_l}|}$. A higher precision leads to more effective song exposure for lesser-known artists.

4.4 Results

We evaluate the performance of the three stakeholders on the following baselines:

- **Popularity:** A naive baseline that always recommend more popular songs from well-known artists, i.e., $\mathcal{R}_l = \emptyset$.
- **Anti-popularity:** A naive baseline that always recommend less popular songs from lesser-known artists, i.e., $\mathcal{R}_w = \emptyset$.
- **WRMF with Likes:** A matrix factorization algorithm trained on likes [10]. It makes the assumption that missing data are negatives.
- **WRMF with Likes and Dislikes:** A matrix factorization algorithms trained on likes and dislikes using the framework proposed in [20].

The Popularity recommender achieves a precision of 41.1% for consumers, while the Anti-popularity recommender results in a lower precision of 36.3%. As a comparison, *WRMF with likes* improve the consumer metric by 21.9% (table 1). Moreover, we find that *WRMF with Likes and Dislikes* outperforms the *WRMF with Likes* for all three stakeholders. The consumer metric was lifted by 18.9%, with an increase of 18.1% for well-known artists and 19.5% for lesser-known artists. This highlights the importance of binary feedback in improving the training and evaluation of recommenders.

5 CONCLUSION

We present the Piki Music Dataset and argue that it was collected in a way that addresses many of the biases of other publicly available datasets. More importantly, since the ratings are binary (in the form of likes and dislikes), we can define performance metrics for recommendation algorithms from the perspective of various stakeholders.

There are a few directions that we think future researchers using this dataset may want to explore. In the spirit of the Netflix challenge, we encourage researchers to test the accuracy of more advanced RecSys algorithms on the dataset. For example, it would be interesting to determine if a neural recommender performs better than matrix factorization for consumers, well-known artists or lesser-known artists. It may also be valuable to explore other metrics to measure

the interests of the stakeholders in this study. Researchers may be also be interested in other ways to segment the artist stakeholders, across genres or other metadata associated with the songs. We are particularly interested in modeling how the algorithm’s objectives are tied to the business objectives of other important stakeholders such as streaming platforms and record labels.

REFERENCES

- [1] Himan Abdollahpouri, Robin Burke, and Masoud Mansoury. 2020. Unfair Exposure of Artists in Music Recommendation. arXiv:2003.11634 [cs.IR]
- [2] Himan Abdollahpouri and Steve Essinger. 2017. Multiple Stakeholders in Music Recommender Systems. arXiv:1708.00120 [cs.CY]
- [3] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. Citeseer, 35.
- [4] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. (2011).
- [5] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. The music streaming sessions dataset. In *The World Wide Web Conference*. 2594–2600.
- [6] Oscar Celma and Pedro Cano. 2008. From hits to niches?: Or how popular artists can bias music recommendation and discovery. *Proc. of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition* (01 2008). <https://doi.org/10.1145/1722149.1722154>
- [7] Andres Ferraro, Xavier Serra, and Christine Bauer. 2021. What is fair? Exploring the artists’ perspective on the fairness of music streaming platforms. arXiv:2106.02415 [cs.HC]
- [8] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. <https://doi.org/10.1145/2827872>
- [9] Robin Burke Ido Guy Dietmar Jannach Toshihiro Kamishima Jan Krasnodebski Himan Abdollahpouri, Gediminas Adomavicius and Luiz Pizzato. 2020. Beyond Personalization: Research Directions in Multistakeholder Recommendation. (2020). <https://arxiv.org/abs/1905.01986>
- [10] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets (*ICDM ’08*). IEEE Computer Society, USA, 263–272. <https://doi.org/10.1109/ICDM.2008.22>
- [11] Dietmar Jannach and Christine Bauer. 2020. Escaping the McNamara Fallacy: Towards more Impactful Recommender Systems Research. *AI Magazine* 41, 4 (2020), 79–95.
- [12] Dominik Kowald, Markus Schedl, and Elisabeth Lex. 2019. The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study. arXiv:1912.04696 [cs.IR]
- [13] Hongyu Lu, Min Zhang, and Shaoping Ma. 2018. Between Clicks and Satisfaction: Study on Multi-Phase User Preferences and Satisfaction for Online News Reading. In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval* (Ann Arbor, MI, USA) (*SIGIR ’18*). Association for Computing Machinery, New York, NY, USA, 435–444. <https://doi.org/10.1145/3209978.3210007>
- [14] Robin Burke Mansoury Masoud and Bamshad Mobasher. 2021. Flatter is better: percentile transformations for recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.2 (2021), 1–16.
- [15] Rishabh Mehrotra, Niannan Xue, and Mounia Lalmas. 2020. Bandit based Optimization of Multiple Objectives on a Music Streaming Platform. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3224–3233.
- [16] Martijn Millecamp, Nyi Nyi Htun, Yucheng Jin, and Katrien Verbert. 2018. Controlling Spotify recommendations: effects of personal characteristics on music recommender user Interfaces. In *Proceedings of the 26th Conference on user modeling, adaptation and personalization*. 101–109.
- [17] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, Vol. 2007. 5–8.
- [18] Yong Li Quanming Yao Chen Gao James Kwok Qiang Yang Isabelle M Guyon Rishabh Mehrotra, Ben A Carterette. 2020. Advances in Recommender Systems: From Multi-stakeholder Marketplaces to Automated RecSys. *KDD ’20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining* (2020), 3533–3534. <https://doi.org/10.1145/3394486.3406463>
- [19] Pablo Sánchez and Alejandro Bellogin. 2018. Measuring Anti-Relevance: A Study on When Recommendation Algorithms Produce Bad Suggestions. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) (*RecSys ’18*). Association for Computing Machinery, New York, NY, USA, 367–371. <https://doi.org/10.1145/3240323.3240382>
- [20] Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging Post-Click Feedback for Content Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) (*RecSys ’19*). Association for Computing Machinery, New York, NY, USA, 278–286. <https://doi.org/10.1145/3298689.3347037>
- [21] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, and Deborah Estrin. 2018. Openrec: A modular framework for extensible and adaptable recommendation algorithms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 664–672.