Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008.

Koren, Y. and Bell, R. Advances in collaborative filtering. *Recommender systems handbook*, pp. 77–118, 2015.

Krauth, K., Dean, S., Zhao, A., Guo, W., Curmei, M., Recht, B., and Jordan, M. I. Do offline metrics predict online performance in recommender systems? *arXiv preprint arXiv:2011.07931*, 2020.

Lukoff, K., Lyngs, U., Zade, H., Liao, J. V., Choi, J., Fan, K., Munson, S. A., and Hiniker, A. How the design of youtube influences user sense of agency. *arXiv preprint arXiv:2101.11778*, 2021.

Mary, J., Gaudel, R., and Preux, P. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pp. 325–336. Springer, 2015.

Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L., and Konstan, J. A. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pp. 677–686, 2014.

Ning, X. and Karypis, G. Slim: Sparse linear methods for top-n recommender systems. In *2011 IEEE 11th International Conference on Data Mining*, pp. 497–506. IEEE, 2011.

Rendle, S. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012. ISSN 2157-6904.

Rendle, S., Zhang, L., and Koren, Y. On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395*, 2019.

Ribeiro, M. H., Ottoni, R., West, R., Almeida, V. A., and Meira Jr, W. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 131–141, 2020.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*, pp. 1670–1679. PMLR, 2016.

Shakespeare, D., Porcaro, L., Gómez, E., and Castillo, C. Exploring artist gender bias in music recommendation. *arXiv preprint arXiv:2009.01715*, 2020.

Singh, A. and Joachims, T. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2219–2228, 2018.

Steck, H. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 154–162, 2018.

Steck, H. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pp. 3251–3257, 2019.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 10–19, 2019.

Wei, Z., Xu, J., Lan, Y., Guo, J., and Cheng, X. Reinforcement learning to rank with markov decision process. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 945–948, 2017.

Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3597–3606, 2020.

Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., and Estrin, D. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 279–287, 2018.

Yao, S., Halpern, Y., Thain, N., Wang, X., Lee, K., Prost, F., Chi, E. H., Chen, J., and Beutel, A. Measuring recommender system effects with simulated users. *arXiv preprint arXiv:2101.04526*, 2021.

Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pp. 337–348. Springer, 2008.

A. Further Examples

Example 3 (Biased MF-SGD)**.** Biased matrix factorization models (Koren & Bell, 2015) compute scores as rating predictions with

$$s_{ui} = \mathbf{p}_u^\top \mathbf{q}_i + f_u + g_i + \mu$$

$P \in \mathbb{R}^{n\times d}$ and $Q \in \mathbb{R}^{m\times d}$ are respectively user and item factors for some latent dimension d, $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{g} \in \mathbb{R}^m$ are respectively user and item biases, and $\mu \in \mathbb{R}$ is a global bias.

The parameters are learned via the regularized optimization

$$\min_{P,Q,\mathbf{f},\mathbf{g},\mu} \frac{1}{2}\sum_u \sum_{i\in\Omega_u} \|\mathbf{p}_u^\top \mathbf{q}_i + f_u + g_i + \mu - r_{ui}\|_2^2 + \frac{\lambda}{2}\|P\|_F^2 + \frac{\lambda}{2}\|Q\|_F^2 \,.$$

Under a stochastic gradient descent minimization scheme (Koren, 2008) with step size α, the one-step update rule for a user factor is

$$\mathbf{p}_u^+ = \mathbf{p}_u - \alpha \sum_{i\in\Omega_u^{\mathcal{A}}} (\mathbf{q}_i\mathbf{q}_i^\top \mathbf{p}_u + \mathbf{q}_i(f_u + g_i + \mu) - \mathbf{q}_i r_{ui}) - \alpha\lambda \mathbf{p}_u \,.$$

User bias terms can be updated in a similar manner, but because the user bias is equal across items, it does not impact the selection of items.

Notice that this expression is affine in the mutable ratings. Therefore, we have an affine score function:

$$\phi_u(\mathbf{a}) = Q\mathbf{p}_u^+ = Q\left((1-\alpha\lambda)\mathbf{p}_u - \alpha Q_{\mathcal{A}}^\top (Q_{\mathcal{A}}\mathbf{p}_u + \mathbf{g}_{\mathcal{A}} + (\mu + f_u)\mathbf{1}) + \alpha Q_{\mathcal{A}}^\top \mathbf{a}\right)$$

where we define $Q_{\mathcal{A}} = Q_{\Omega_u^{\mathcal{A}}} \in \mathbb{R}^{|\Omega_u^{\mathcal{A}}|\times d}$ and $\mathbf{g}_{\mathcal{A}} = \mathbf{g}_{\Omega_u^{\mathcal{A}}} \in \mathbb{R}^{|\Omega_u^{\mathcal{A}}|}$. Therefore,

$$B_u = \alpha Q Q_{\mathcal{A}}^\top, \quad \mathbf{c}_u = Q\left((1+\lambda)\mathbf{p}_u - \alpha Q_{\mathcal{A}}^\top (Q_{\mathcal{A}}\mathbf{p}_u + \mathbf{g}_{\mathcal{A}} + (\mu + f_u)\mathbf{1})\right) \,.$$

Example 4 (Biased MF-ALS)**.** Rather than a stochastic gradient descent minimization scheme, we may instead update the model with an alternating least-squares strategy (Zhou et al., 2008). In this case, the update rule is

$$\begin{aligned}\mathbf{p}_u^+ &= \arg\min_{\mathbf{p}} \sum_{i\in\Omega_u^{\mathcal{A}}\cap\Omega_u} \|\mathbf{p}^\top \mathbf{q}_i + f_u + g_i + \mu - r_{ui}\|_2^2 + \lambda\|\mathbf{p}\|_2^2 \\ &= (Q_u^\top Q_u + \lambda I)^{-1}(Q^\top \mathbf{r}_u + Q_{\mathcal{A}}^\top(\mathbf{g}_{\mathcal{A}} + (\mu + f_u)\mathbf{1}) + Q_{\mathcal{A}}^\top \mathbf{a})\end{aligned}$$

where we define $Q_u = Q_{\Omega_u^{\mathcal{A}}\cap\Omega_u}$. Similar to in the SGD setting, this is an affine expression, and therefore we end up with the affine score parameters

$$B_u = Q(Q_u^\top Q_u + \lambda I)^{-1}Q_{\mathcal{A}}^\top, \quad \mathbf{c}_u = Q(Q_u^\top Q_u + \lambda I)^{-1}(Q^\top \mathbf{r}_u + Q_{\mathcal{A}}^\top(\mathbf{g}_{\mathcal{A}} + (\mu + f_u)\mathbf{1})) \,.$$

Example 5 (Biased Item-KNN)**.** Biased neighborhood models (Desrosiers & Karypis, 2011) compute scores as rating predictions by a weighted average, with

$$s_{ui} = \mu + f_u + g_i + \frac{\sum_{j\in\mathcal{N}_i} w_{ij}(r_{uj} - \mu - f_u - g_i)}{\sum_{j\in\mathcal{N}_i} |w_{ij}|}$$

where w_{ij} are weights representing similarities between items, $\mathcal{N}_i$ is a set of indices which are in the neighborhood of item i, and $\mathbf{f}, \mathbf{g}, \mu$ are bias terms. Regardless of the details of how these parameters are computed, the predicted scores are an affine function of observed scores:

$$\mathbf{s}_u = W\mathbf{r}_u - W(\mathbf{g} + (\mu + f_u)\mathbf{1}) + \mathbf{g} + (\mu + f_u)\mathbf{1}$$

where we can define

$$W_{ij} = \begin{cases} \frac{w_{ij}}{\sum_{j\in\mathcal{N}_i}|w_{ij}|} & j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the score updates take the form

$$\phi_u(\mathbf{a}) = \underbrace{W(\mathbf{r}_u - \mathbf{g} + (\mu + f_u)\mathbf{1}) + \mathbf{g} + (\mu + f_u)\mathbf{1})}_{\mathbf{c}_u} + \underbrace{W E_{\Omega_u^{\mathcal{A}}}}_{B_u} \mathbf{a}$$

where $E_{\Omega_u^{\mathcal{A}}}$ selects rows of W corresponding to action items.

Example 6 (SLIM and EASE)**.** For both SLIM (Ning & Karypis, 2011) and EASE (Steck, 2019), scores are computed as

$$s_{ui} = \mathbf{w}_i^\top \mathbf{r}_u$$

for $\mathbf{w}_i$ the row vectors of a weight matrix W. For SLIM, the sparse weights are computed as

$$\begin{aligned} \min_W \ & \frac{1}{2}\|R - RW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1 \\ \text{s.t. } & W \geq 0, \operatorname{diag}(W) = 0 \end{aligned}$$

For EASE, the weights are computed as

$$\begin{aligned} \min_W \ & \frac{1}{2}\|R - RW\|_F^2 + \lambda\|W\|_F^2 \\ \text{s.t. } & \operatorname{diag}(W) = 0 \end{aligned}$$

In both cases, the score updates take the form

$$\phi_u(\mathbf{a}) = \underbrace{W\mathbf{r}_u}_{\mathbf{c}_u} + \underbrace{W E_{\Omega_u^{\mathcal{A}}}}_{B_u} \mathbf{a}\,.$$

B. Proofs of Results

Proof of Proposition 1. Define

$$\gamma_\beta(\mathbf{a}) = \operatorname*{LSE}_{j\in\Omega_u^t}\left(\beta\phi_{uj}(\mathbf{a})\right) - \beta\phi_{ui}(\mathbf{a})$$

and see that $\rho_{ui}(\mathbf{a}) = e^{-\gamma_\beta(\mathbf{a})}$. Then we see that

$$\lim_{\beta\to\infty} \frac{1}{\beta}\gamma_\beta(\mathbf{a}) = \max_{j\notin\Omega_u}\left(\phi_{uj}(\mathbf{a})\right) - \phi_{ui}(\mathbf{a})$$

yields a top-1 expression. If an item i is top-1 reachable for user u, then there is some $\mathbf{a}$ such that the above expression is equal to zero. Therefore, as $\beta \to \infty$, $\gamma^\star \to 0$, hence $\rho^\star \to 1$. In the opposite case when an item *is not* top-1 reachable we have that $\gamma^\star \to \infty$, hence $\rho^\star \to 0$. □

Definition 2 (Convex hull)**.** The *convex hull* of a set of vectors $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^n$ is defined as

$$\operatorname{conv}(\mathcal{V}) = \left\{ \sum_{i=1}^n w_i \mathbf{v}_i \mid \mathbf{w} \in \mathbb{R}_+^n, \ \ \sum_{i=1}^n w_i = 1 \right\}.$$

A point $\mathbf{v}_j \in \mathcal{V}$ is a *vertex* of the convex hull if

$$\mathbf{v}_j \notin \operatorname{conv}\left(\mathcal{V} \setminus \{\mathbf{v}_j\}\right).$$

Proof of Proposition 2. We begin by showing that if $\mathbf{b}_{ui}$ is a vertex on the convex hull of $\mathcal{B} = \{\mathbf{b}_{uj}\}_{j\in\Omega_u^t}$, then item i is top-1 reachable. This argument is similar to the proof of Results 1 and 2 in (Dean et al., 2020).

Item i is top-1 reachable if there exists some $\mathbf{a} \in \mathbb{R}^{|\Omega_u^{\mathcal{A}}|}$ such that $\mathbf{b}_{ui}^\top\mathbf{a} + c_{ui} \geq \mathbf{b}_{uj}^\top\mathbf{a} + c_{uj}$ for all $j \neq i$. Therefore, top-1 reachability is equivalent to the feasibility of the following linear program

$$\begin{aligned} \min \ & 0^\top \mathbf{a} \\ \text{s.t. } & D_{ui}\mathbf{a} \geq \mathbf{f}_{ui} \end{aligned}$$

where D_{ui} has rows given by $\mathbf{b}_{ui} - \mathbf{b}_{uj}$ and $\mathbf{f}_{ui}$ has entries given by $c_{uj} - c_{ui}$ for all $j \in \Omega_u^t$ with $j \neq i$. Feasibility of this linear program is equivalent to boundedness of its dual:

$$\begin{aligned} &\max \mathbf{f}_{ui}^\top \lambda \\ &\text{s.t. } D_{ui}^\top \lambda = 0, \;\; \lambda \geq 0. \end{aligned}$$

We now show that if $\mathbf{b}_{ui}$ is a vertex on the convex hull of $\mathcal{B}$, then the dual is bounded because the only feasible solution is $\lambda = 0$. To see why, notice that

$$D_{ui}^\top \lambda = 0 \iff \mathbf{b}_{ui} \sum_{\substack{j \in \Omega_u^t \\ j \neq i}} \lambda_j = \sum_{\substack{j \in \Omega_u^t \\ j \neq i}} \lambda_j \mathbf{b}_{uj}$$

If this expression is true for some $\lambda \neq 0$, then we can write

$$\mathbf{b}_{ui} = \sum_{\substack{j \in \Omega_u^t \\ j \neq i}} w_j \mathbf{b}_{uj}, \quad w_j = \frac{\lambda_j}{\sum_{\substack{j \in \Omega_u^t \\ j \neq i}} \lambda_j} \implies \mathbf{b}_{ui} \in \operatorname{conv}\left(\mathcal{B} \setminus \{\mathbf{b}_{ui}\}\right) \, .$$

This is a contradiction, and therefore it must be that $\lambda = 0$ and therefore the dual is bounded and item i is top-1 reachable.

To finish the proof, we appeal to Proposition 1 to argue that since item i is top-1 reachable, then $\rho_{ui}^\star \to 1$ as $\beta \to \infty$. $\square$

Definition 3 (Rich actions)**.** For a set of item factors $\{\mathbf{q}_j\}_{j=1}^m$, let $C = \max_j \|\mathbf{q}_j\|_2$. Then a set of action items $\Omega_u^{\mathcal{A}} \subseteq \{1, \dots, m\}$ is *sufficiently rich* if the vertical concatenation of their item factors and norms is full rank:

$$\operatorname{rank}\Big(\begin{bmatrix} \mathbf{q}_i^\top & \sqrt{C^2 - \|\mathbf{q}_i\|_2^2} \end{bmatrix}_{i \in \Omega_u^{\mathcal{A}}} \Big) = d + 1 \, .$$

Notice that this can only be true if $|\Omega_u^{\mathcal{A}}| \geq d + 1$.

Proof of Proposition 3. Let C be the maximum row norm of Q and define $\mathbf{v} \in \mathbb{R}^m$ satisfying $v_i^2 = C^2 - \|\mathbf{q}_i\|_2^2$. Then we construct modified item and user factors as

$$\tilde{Q} = \begin{bmatrix} Q & \mathbf{v} \end{bmatrix}, \quad \tilde{P} = \begin{bmatrix} P & \mathbf{0} \end{bmatrix} \, .$$

Therefore, we have that $\tilde{P}\tilde{Q}^\top = PQ^\top$.

Then notice that by construction, each row of $\tilde{Q}$ has norm C, so each $\tilde{\mathbf{q}}_i$ is on the boundary of the ℓ_2 ball in $\mathbb{R}^{d+1}$. As a result, each $\tilde{\mathbf{q}}_i$ is a vertex on the convex hull of $\{\tilde{\mathbf{q}}_j\}_{j=1}^n$ as long as all $\mathbf{q}_j$ are unique.

For an arbitrary user u, the score model parameters are given by $\tilde{\mathbf{b}}_{ui} = \tilde{Q}_{\mathcal{A}} \tilde{\mathbf{q}}_i$. We show by contradiction that as long as the action items are sufficiently rich, each $\tilde{\mathbf{b}}_{ui}$ is a vertex on the convex hull of $\{\tilde{\mathbf{b}}_{uj}\}_{j=1}^n$. Supposing this is not the case for an arbitrary i,

$$\tilde{\mathbf{b}}_{ui} = \sum_{\substack{j=1 \\ j \neq i}}^n w_j \tilde{\mathbf{b}}_{uj} \iff \tilde{Q}_{\mathcal{A}} \tilde{\mathbf{q}}_i = \sum_{\substack{j=1 \\ j \neq i}}^n w_j \tilde{Q}_{\mathcal{A}} \tilde{\mathbf{q}}_i \implies \tilde{\mathbf{q}}_i = \sum_{\substack{j=1 \\ j \neq i}}^n w_j \tilde{\mathbf{q}}_i$$

where the final implication follows because the fact that $\tilde{Q}_{\mathcal{A}}$ is full rank (due to richness) implies that $\tilde{Q}_{\mathcal{A}}^\top \tilde{Q}_{\mathcal{A}}$ is invertible. This is a contradiction, and therefore we have that each $\tilde{\mathbf{b}}_{ui}$ must be a vertex on the convex hull of $\{\tilde{\mathbf{b}}_{uj}\}_{j=1}^n$.

Finally, we appeal to Proposition 2 to argue that $\rho^\star(u, i) \to 1$ as $\beta \to \infty$ for all target items $i \in \Omega_u^t$. $\square$

C. Datasets, Model Training and Computing Infrastructure

C.1. Detailed data description

Table 1 provides summary statistics.

Table 1. Audit datasets

Data set	ML 1M	LastFM 360K	MIND
Users	6040	13698	50000
Items	3706	20109	247
Ratings	1000209	178388	670773
Density (%)	4.47%	0.065%	5.54%
LibFM RMSE	0.716	1.122	0.318
KNN RMSE	0.756	1.868	-

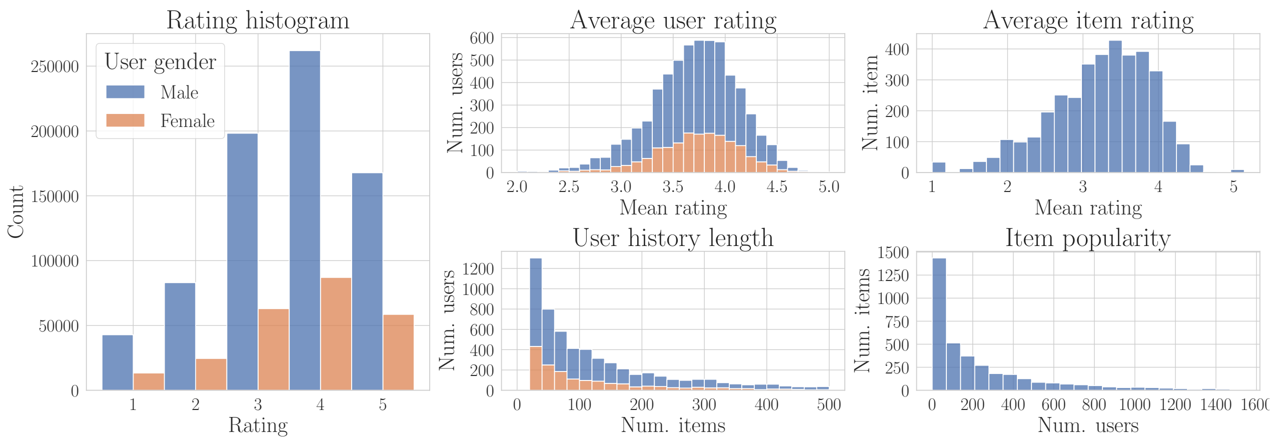

Figure 10. Descriptive statistics for the MovieLens 1M dataset split by user gender (28.3% female). The mean ratings of both users and items are roughly normally distributed while user's history length and item popularity display power law distributions.

MovieLens 1 Million ML-1M dataset was downloaded from Group Lens[2] via the RecLab (Krauth et al., 2020) interface[3]. It contains 1 through 5 rating data of 6040 users for 3706 movies. There are a total of 1000209 ratings (4.47% rating density). The original data is accompanied by additional user attributes such as age, gender, occupation and zip code. Our experiments didn't indicate observable biases across these attributes. In Section E we show user discovery results split by gender.

Figure 10 illustrates descriptive statistics for the ML-1M dataset.

LastFM 360K The LastFM 360K dataset preprocessed[4] by Shakespeare et al. (2020) was loaded via the RecLab interface. It contains data on the number of times users have listened to various artists. We select a random subset of 10% users and a random subset of 10% items yielding 13698 users, 20109 items and 178388 ratings (0.056% rating density). The item ratings are not explicitly expressed by users as in the MovieLens case. For a user u and an artist i we define implicit ratings $r_{ui} = \log(\#\text{listens}(u,i) + 1)$. This data is accompanied by artist gender, an item attribute.

Figure 11 illustrates descriptive statistics for the LastFM dataset.

MIcrosoft News Dataset (MIND) MIND is a recently published impression dataset collected from logs of the Microsoft News website [5]. We downloaded the `MIND-small` dataset[6], which contains behaviour log data for 50000 randomly sampled users. There are 42416 unique news articles, spanning 17 categories and 247 subcategories. We aggregate user interactions at the subcategory level and consider the problem of news subcategory recommendation. The implicit rating of a user u for subcategory i is defined as: $r_{ui} = \log(\#\text{clicks}(u,i) + 1)$. The resulting aggregated dataset contains 670773 ratings (5.54% rating density).

[2] https://grouplens.org/datasets/movielens/1m/
[3] https://github.com/berkeley-reclab/RecLab
[4] https://zenodo.org/record/3964506#.XyE5N0FKg5n
[5] https://microsoftnews.msn.com/
[6] https://msnews.github.io/

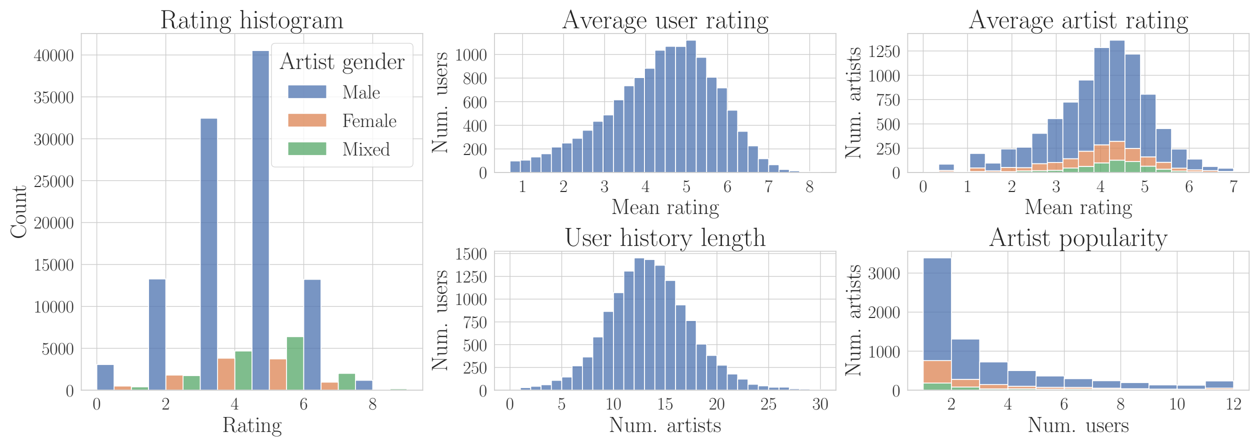

Figure 11. Descriptive statistics for the LastFM dataset split by artist gender (over 54% of artists have unknown gender, 36% are male, 6.5% are female and 3.5% are mixed gender). Unlike ML 1M, for the LastFM dataset the user history lengths are normally distributed around a mean of around 12 artists.

Table 2. Tuning results

DATASET	LIBFM LR	LIBFM REG.	LIBFM TEST RMSE	LIBFM RUN TIME (S)	KNN NEIGH. SIZE	KNN SHRINKAGE	KNN TEST RMSE	KNN RUN TIME (S)
ML 1M	0.0112	0.0681	0.716	2.76 ± 0.32	100	22.22	0.756	0.34 ± 0.07
LASTFM	0.0478	0.2278	1.122	0.78 ± 0.13	-	-	-	-
MIND	0.09	0.0373	0.318	3.23 ± 0.37	-	-	-	-

Figure 12 illustrates descriptive statistics for the MIND dataset.

C.2. Model Tuning

For each dataset and recommender model we perform grid search for progressively finer meshes over the tunable hyper-parameters of the recommender. We use recommenders implemented by the RecLab library. For each dataset and recommender we evaluate hyperparameters on a 10% split of test data. The best hyper-parameters for each setting are presented in Table 2.

LibFM We performed hyper-parameter tuning to find suitable learning rate and regularization parameter for each dataset. Following (Dacrema et al., 2021) we consider $\texttt{lr} \in (0.001, 0.5)$ as the range of hyper-parameters for the learning rate and $\texttt{reg} \in (10^{-5}, 10^{0})$ for the regularization parameter. In all experimental settings we follow the setup of (Rendle et al., 2019) and use 64 latent dimensions and train with SGD for 128 iterations.

KNN We perform hyperparameter tuning with respect to neighborhood size and shrinkage parameter. Following (Dacrema et al., 2021) we consider the range $(5, 1000)$ for the neighborhood size and $(0, 1000)$ for the shrinkage parameter. We tune KNN only for the ML-1M dataset.

C.3. Experimental Infrastructure and Computational Complexity

All experiments were performed on a 64 bit desktop machine equipped with 20 CPUs (Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz) and a 62 GiB RAM. Average run times for training an instance of each recommender can be found in Table 2.

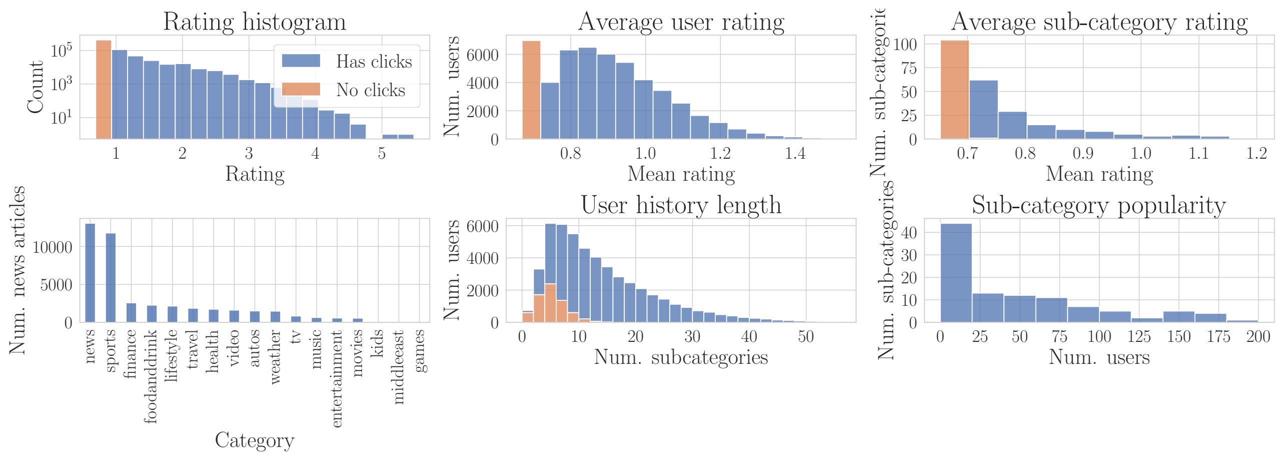

Figure 12. Descriptive statistics for the MIND dataset: The orange bars correspond to either user or items that have been displayed but have not clicked/ have not been clicked on. Unlike ML 1M and LastFM, the MIND ratings have strongly skewed distribution, with most user-subcategory ratings corresponding to users clicking on a small number of articles from the sub-category. There is a long tail of higher ratings that corresponds to most popular subcategories. The leftmost plot illustrates the unequal distribution of news articles across categories. The same qualitative behaviour holds for sub-categories.

D. Computing Reachability

D.1. Conic Program Implementation

The optimization problem in (4) is convex, and we solve it as a conic optimization problem using the MOSEK Python API under an academic license (ApS, 2019). We reformulate (4) as an optimization over the exponential cone:

$$\begin{aligned}\min_{t,\mathbf{a},\mathbf{u}} \quad & t-\beta(\mathbf{b}_{ui}^\top\mathbf{a}+c_{ui})\\ \text{s.t.} \quad & \mathbf{a}\in\mathcal{A}_u, \quad \sum_{j\in\Omega_u^t} u_j\le 1,\\ & \big(u_j,1,\beta(\mathbf{b}_{uj}^\top\mathbf{a}+c_{uj})-t\big)\in\mathcal{K}_{exp} \;\;\forall\;\, j\in\Omega_u^t\end{aligned} \tag{5}$$

The parameters B_u and $\mathbf{c}_u$ are computed for each user based on the recommender model as described in Section A. For the LibFM model, we consider user updates with $\alpha=0.1$ and $\lambda=0$. Average run times for computing reachability of a user-item pair in various settings can be found in Table 3.

D.2. Experimental Setup for Computing Reachability

ML 1M We compute max stochastic reachability for the LibFM and KNN preference model. We consider three types of user action spaces: *History Edits*, *Future Edits*, and *Next K* in which users can strategically modify the ratings associated to K randomly chosen items from their history, K randomly chosen items from that they have not yet seen, or the top-K unseen items according to the baseline scores of the preference model. For each of the action spaces we consider $K\in\{5,10,20\}$.

We perform reachability experiments on a random 3% subset of users (176). For each choice of preference model, action space type and action space size we sample for each user 500 random items that have not been previously rated and are not action items. For each user-item pair we compute reachability for a range of stochasticity parameters $\beta\in\{1,2,4,10\}$. Note that across all experimental settings we compute reachability for the same subset of users, but different subsets of randomly selected target items.

We use the ML 1M dataset to primarily gain insights in the role that preference models, item selection stochasticity and strategic action spaces play in determining the maximum achievable degree of stochastic reachability in a recommender system.

Table 3. Reachability run times (in seconds).

NUM. ACTIONS	ML 1M (LIBFM)	ML 1M (KNN)	LASTFM	MIND
K = 5	0.82 ± 0.04	9.8 ± 3.4	-	-
K = 10	0.87 ± 0.04	10.2 ± 6.1	4.91 ± 0.32	0.44 ± 0.01
K= 20	0.91 ± 0.05	11.4 ± 6.8	-	-

LastFM We run reachability experiment for LibFM recommender with *Next K = 10* action model and stochasticity parameter $\beta = 2$. We compute $\rho^\star$ values for 100 randomly sampled users and 500 randomly sampled items from the set of non-action items (target items can include previously seen items). Unlike the ML 1M dataset, the set of target items is shared among all users.

MIND We run reachability experiments for LibFM recommender with *Next K = 10* action model and stochasticity parameter $\beta = 2$. We compute reachability for all items and users.

Reachability Run Times In Table 3 we present the average clock time for computing reachability for a user-item pair in the settings described above. Due to internal representation of action spaces as matrices the runtime dependence on the dimension of the action space is fairly modest. We do not observe significant run time differences between different types of action spaces. We further add multiprocessing functionality to parallelize reachability computations over multiple target items.

E. Detailed Experimental Results

E.1. Impact of recommender design

We present further insights in the experimental settings studied in Section 5.2. For ML-1M, we replicate the log scale scatterplots of $\rho^\star$ against baseline ρ for all the action spaces (*Next K, Random Future, Random History*), the full range of $\beta \in \{1, 2, 4, 10\}$ and the two preference models: LibFM (Figure 13) and KNN (Figure 14). We observe that for both KNN and LibFM, random history edits can lead to higher $\rho^\star$ values. We posit that this increased agency is partly due to the fact that when editing K items from the history a user edits a larger fraction of total ratings compared to editing K future items.

The most striking feature of KNN reachability results is the strong correlation between baseline ρ and $\rho^\star$. The correlations between baseline and max probability of recommendation is less strong in the case of LibFM. These insights are corroborated by Figure 15 which compares the average LibFM and KNN user lifts for different choices of action space, action size *K*, stochasticity parameter β.

E.2. Bias in movie, music, and news recommendation

We present further results on the settings studied in Section 5.3. We replicate the popularity bias results on ML-1M for different action spaces and plot the results in Figure 16. We see that the availability bias for KNN is dependent on the action space, with *Random History* displaying no or little correlation between popularity and max availability. This is not surprising given the results in Figure 6.

To systematically study the popularity bias, we compute the Spearman rank-order correlation coefficient to measure the presence of a monotonic relationship between popularity (as measured by average rating) and availability (either in the baseline or max case). We also compute the correlation between the popularity and the prevalence in the dataset, as measured by number of ratings.

The impact of user action spaces is displayed in Figure 17, which plots the correlation between popularity and max availability for different action spaces. For comparison, the correlation between popularity and baseline availability is just over 0.8 for all of these settings[7], while the correlation with dataset prevalence is 0.346. Table 4 shows these correlation values across datasets for a fixed action model. In all cases with the LibFM model, the pattern that popularity is less correlated with max availability than baseline availability holds; however, the correlation with dataset prevalence varies.

[7]Due to variation in baseline actions, the baseline availability is not exactly the same.

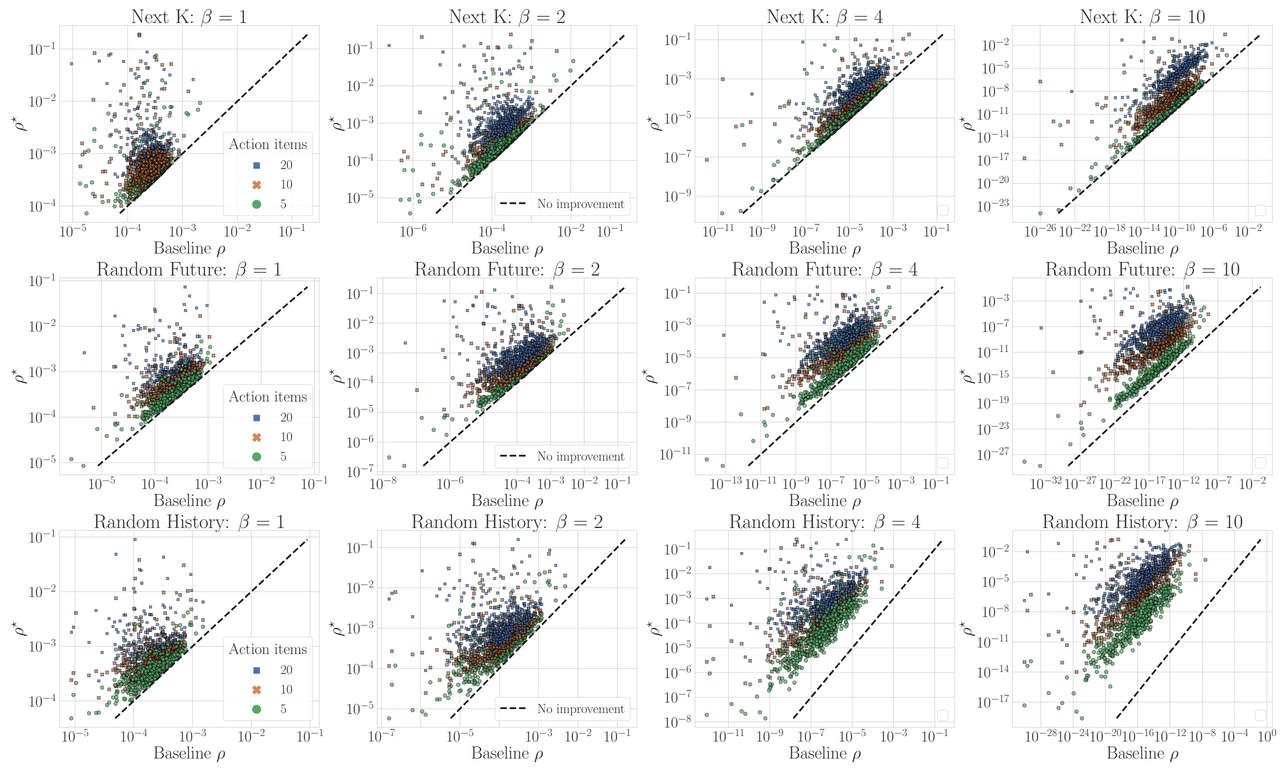

Figure 13. Log scale scatterplots of $\rho^\star$ against baseline ρ evaluated for the LibFM preference model.

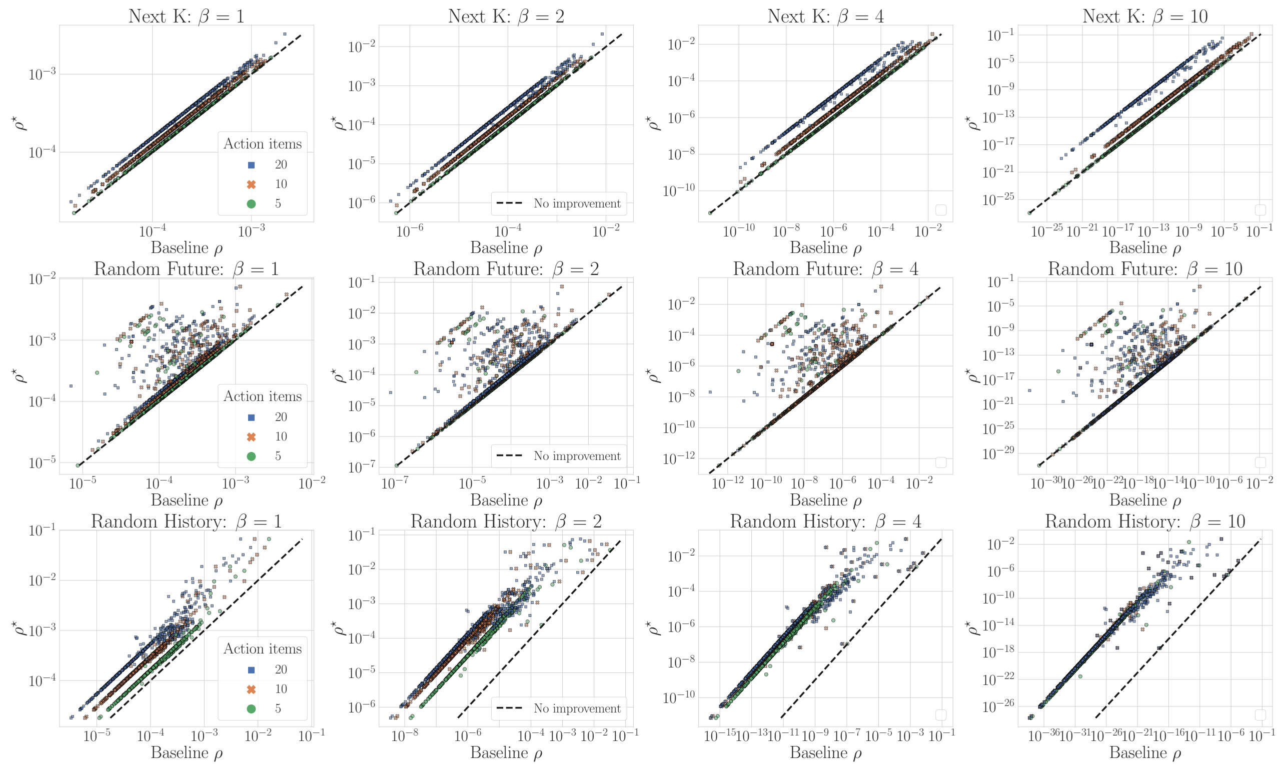

Figure 14. Log scale scatterplots of $\rho^\star$ against baseline ρ evaluated for the KNN preference model.

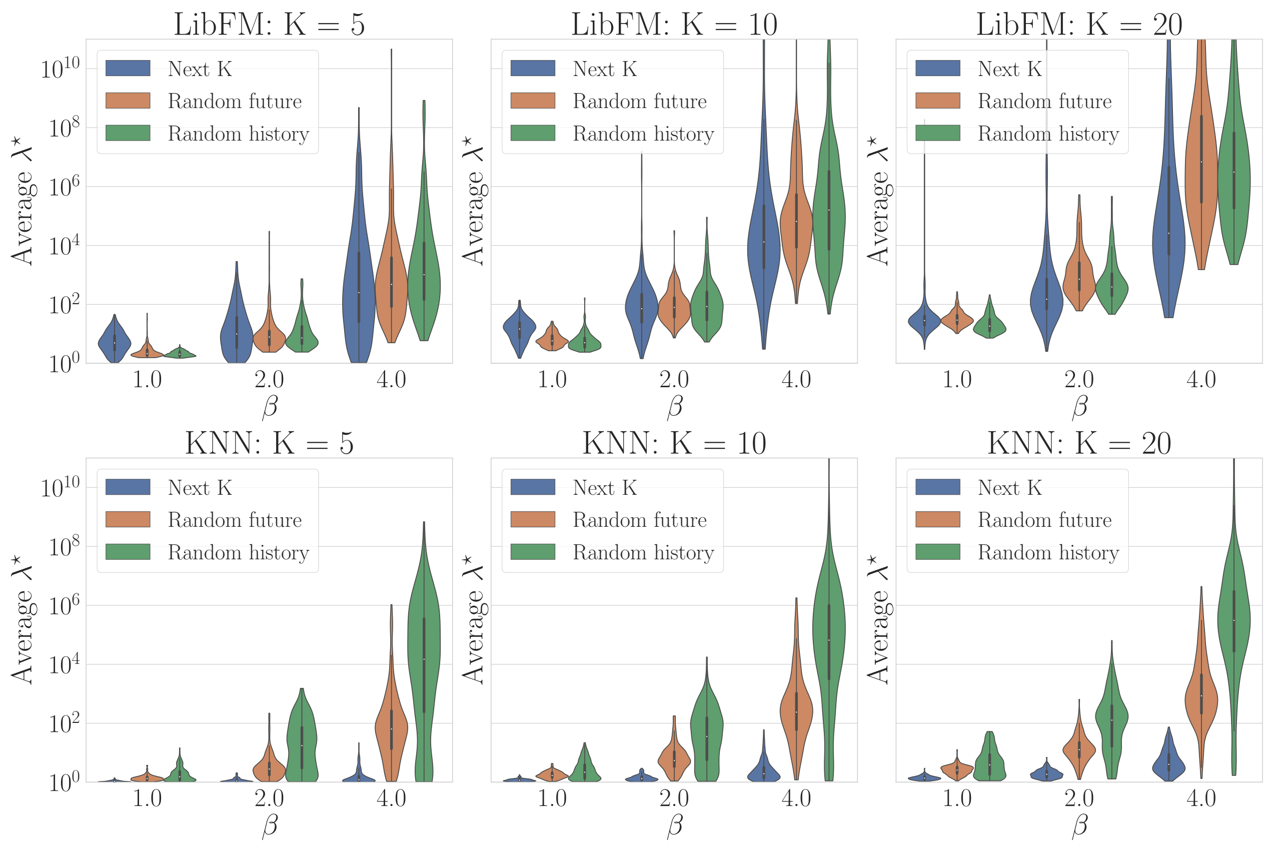

Figure 15. Side by side comparison of average user lifts for LibFM (top row)and KNN(bottom row).

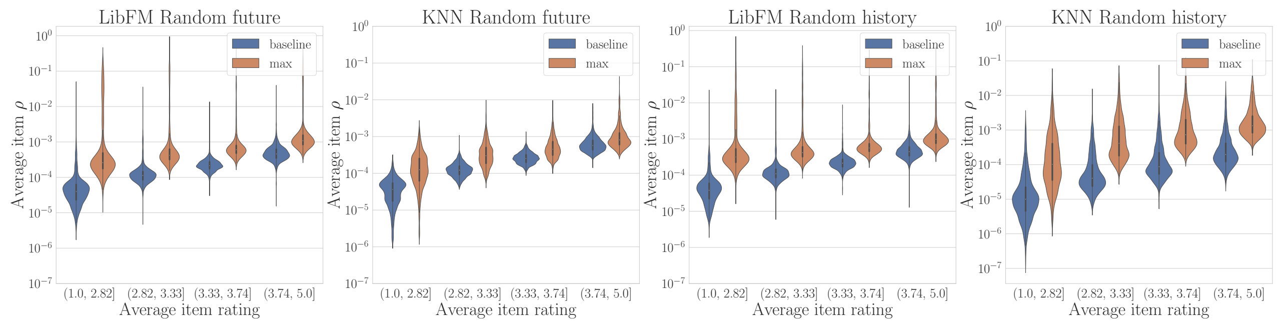

Figure 16. Side by side comparison of baseline and best-case availability of content, across four popularity categories. From left to right: LibFM preference model with *Random Future*, KNN preference model with *Random Future*, LibFM preference model with *Random History*, KNN preference model with *Random History*. Reachability evaluated on ML-1M for with $K = 10$ and $\beta = 2$.

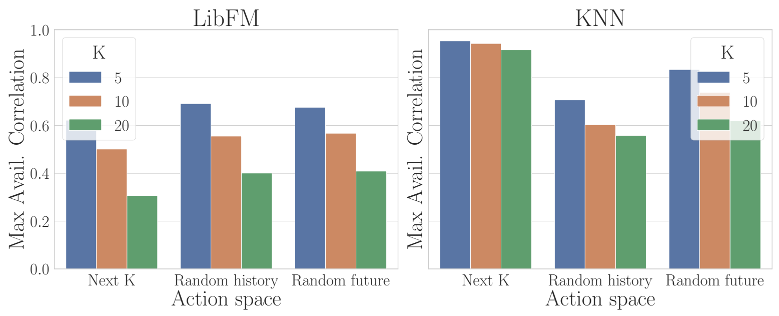

Figure 17. Comparison of Spearman's correlation between item popularity and max availability for different action spaces and models. Reachability evaluated on ML-1M with $\beta = 2$.

Table 4. Spearman's correlation with popularity for *Next K* with $K = 10$ and $\beta = 2$.

DATASET	MODEL	CORR. WITH DATASET PREVALENCE	CORR. WITH BASELINE AVAILABILITY	CORR. WITH MAX AVAILABILITY
ML-1M	LIBFM	0.346280	0.827492	0.501316
ML-1M	KNN	0.346280	0.949581	0.942986
MIND	LIBFM	0.863992	0.825251	0.435212
LASTFM	LIBFM	0.133318	0.671101	0.145949

To investigate experience bias, we similarly compute the Spearman rank-order correlation coefficient to measure the presence of a monotonic relationship between user experience (as measured by number of items rated) and discovery (either in the baseline or max case). We observe correlation values of varying sign across datasets and models, and none are particularly strong (Table 5).

Finally, we investigate gender bias. We compare discovery across user gender for ML-1M and availability across artist gender for LastFM (Figure 18). We do not observe any trends in either baseline or max values.

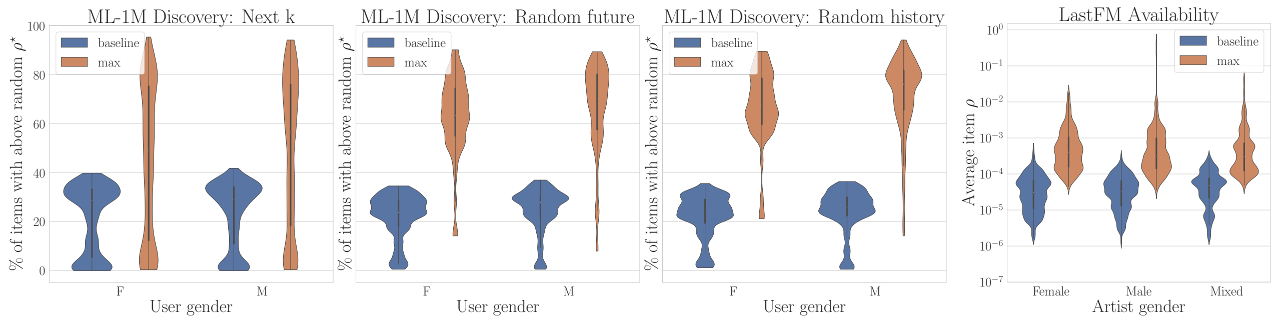

Figure 18. Side by side comparison of baseline and maximum discovery across user gender (left 3 panels) and availability across artist gender (rightmost panel). Reachability evaluated on ML-1M and LastFM with LibFM model, $K = 10$, different action spaces, and $\beta = 2$.

Table 5. Spearman's correlation with experience for *Next K* with $K = 10$ and $\beta = 2$.

DATASET	MODEL	CORR. WITH BASELINE DISCOVERY	CORR. WITH MAX DISCOVERY
ML-1M	LIBFM	0.475777	0.530359
ML-1M	KNN	0.206556	-0.031929
MIND	LIBFM	0.050961	0.112558
LASTFM	LIBFM	-0.084130	-0.089226