# On Item-Sampling Evaluation for Recommender System

DONG LI and RUOMING JIN, Kent State University, USA
ZHENMING LIU and BIN REN, College of William & Mary, USA
JING GAO and ZHI LIU, iLambda Inc., USA

Personalized recommender systems play a crucial role in modern society, especially in e-commerce, news, and ads areas. Correctly evaluating and comparing candidate recommendation models is as essential as constructing ones. The common offline evaluation strategy is holding out some user-interacted items from training data and evaluating the performance of recommendation models based on how many items they can retrieve. Specifically, for any hold-out item or so-called target item for a user, the recommendation models try to predict the probability that the user would interact with the item and rank it among overall items, which is called *global evaluation*. Intuitively, a good recommendation model would assign high probabilities to such hold-out/target items. Based on the specific ranks, some metrics like *Recall@K* and *NDCG@K* can be calculated to further quantify the quality of the recommender model. Instead of ranking the target items among all items, Koren first proposed to rank them among a small *sampled set of items*, then quantified the performance of the models, which is called *sampling evaluation*. Ever since then, there has been a large amount of work adopting sampling evaluation due to its efficiency and frugality. In recent work, Rendle and Krichene argued that the sampling evaluation is "inconsistent" with respect to a global evaluation in terms of offline top-$K$ metrics.

In this work, we first investigate the "inconsistent" phenomenon by taking a glance at the connections between sampling evaluation and global evaluation. We reveal the approximately linear relationship between sampling with respect to its global counterpart in terms of the top-$K$ Recall metric. Second, we propose a new statistical perspective of the sampling evaluation—to estimate the global rank distribution of the entire population. After the estimated rank distribution is obtained, the approximation of the global metric can be further derived. Third, we extend the work of Krichene and Rendle, directly optimizing the error with ground truth, providing not only a comprehensive empirical study but also a rigorous theoretical understanding of the proposed metric estimators. To address the "blind spot" issue, where accurately estimating metrics for small top-$K$ values in sampling evaluation is challenging, we propose a novel adaptive sampling method that generalizes the expectation-maximization algorithm to this setting. Last but not least, we also study the user sampling evaluation effect. This series of works outlines a clear roadmap for sampling evaluation and establishes a foundational theoretical framework. Extensive empirical studies validate the reliability of the sampling methods presented.

CCS Concepts: • **Information systems** → **Presentation of retrieval results**; **Collaborative filtering**; **Personalization**; **Content ranking;**

Additional Key Words and Phrases: Recommender system, sampling evaluation, top-$K$ metric

This research was partially funded by the National Science Foundation through grants IIS-2142675, IIS-2142681, and III-2008557. Additional support came from a collaborative research agreement between Kent State University and iLambda Inc.
Authors' addresses: D. Li and R. Jin, Kent State University, 800 E Summit St, Kent, 44240, OH; e-mails: {dli12, rjin1}@kent.edu; Z. Liu and B. Ren, College of William & Mary, 200 Stadium Dr, Williamsburg, 23185, VA; e-mails: {zliu, bren}@cs.wm.edu; J. Gao and Z. Liu, iLambda Inc., 251 W Garfield Rd, Suite 150 Aurora, 44202, OH; e-mails: {jgao, zliu}@ilambda.com.

2770-6699/2024/03-ART7 $15.00
https://doi.org/10.1145/3629171

**ACM Reference format:**
Dong Li, Ruoming Jin, Zhenming Liu, Bin Ren, Jing Gao, and Zhi Liu. 2024. On Item-Sampling Evaluation for Recommender System. *ACM Trans. Recomm. Syst.* 2, 1, Article 7 (March 2024), 36 pages.
https://doi.org/10.1145/3629171

## 1 INTRODUCTION

Recommender systems have become a vital and integral part of modern lives, transforming user experiences across various sectors such as e-commerce, online advertising, streaming, and social media platforms [15, 19, 33, 34, 39]. By analyzing users' profiles, preferences, and historical data, recommender systems offer personalized suggestions tailored to individual interests and needs. Their capacity to process massive amounts of information helps users navigate the abundance of available content and discover relevant products, services, or materials. Consequently, recommender systems not only enhance user engagement and loyalty, providing them with a better experience, but also contribute to increased sales and revenue for businesses, such as Amazon and YouTube. Thus, recommendation, or personalization, which aims to best match the preferences and/or needs of an individual customer across all available choices, is simply indispensable.

As personalization and recommendation continue to play an integral role in the emerging AI-driven economy [15, 21, 30, 39], proper and rigorous evaluation of recommendation models has become increasingly important in recent years for both academic researchers and industry practitioners [4, 6, 7, 13, 32]. Thanks to the widely available and ever-increasing recommendation models [12, 39], data scientists today spend significant amounts of time evaluating, deploying, testing, and fine-tuning recommendation models. Online A/B tests are the ultimate criteria for discerning different recommendation models; however, running such a test often takes days or even weeks to draw a conclusion. Offline evaluations thus play a critical role in helping to choose promising/right recommendation models for online testing. Offline evaluation typically holds out some interacted items from the training data. Specifically, for any hold-out items or so-called target items for a user, the recommendation models attempt to predict the probability of user-item interaction and rank the item among all items, which is referred to as *global evaluation*. Intuitively, an effective recommendation model would assign high probabilities to such hold-out/target items, and based on their specific ranks, metrics like *Recall@K* and *NDCG@K* can be calculated to further quantify the model's quality.

***History on Sampling Top-K Evaluation.*** Contrary to *global evaluation*, Koren [22], for the first time, used the sampling top-$K$ method in his seminal work as an approach to measure the success of top-$K$ recommenders. Specifically, he uses $1,000$ additional random movies (which may include already-ranked ones) against the targeted movie $i$ for a user. He ranks these $1,001$ movies by the predicted rating (relevance score), and he normalizes the ranking score between 0 and 1. Finally, he draws the cumulative distributions of all users, with respect to the ranking score. In summary, ranking the target item among a small sampled set of items, and then quantifying the performance of the models, is called *item-sampling-based evaluation*. Another highly cited work [5] has utilized this metric to evaluate the performance of a variety of recommendation algorithms on top-$N$ recommendation tasks. This method was first adopted by deep learning based recommendation papers in 2015 [11] and then in 2017 [16]. Here, the authors go beyond the top-$K$ Hit-Ratio suggested by Koren [11, 16], extending to metrics such as Mean Reciprocal Rank (MRR) and NDCG. Since Koren, various deep learning based recommendation studies [10, 17, 23, 36–38] have adopted such sampling-based top-$K$ evaluation metrics. In these studies, they typically sample only those "irrelevant" items (not scored by the users), unlike the work in Koren, which may sample relevant items, as well. The number of items sampled typically ranges from 100 to 1,000.

***Inconsistency of Sampled Metrics.*** Despite the popularity of sampling-based evaluation [10, 16, 17, 23, 36–38], recently, Rendle [32] and Krichene and Rendle [24] argued that sampling-based top-$K$ evaluation metrics, such as Recall/Precision (Hit-Ratio), **Average Precision (AP)**, and NDCG, excluding AUC, are "inconsistent" with global metrics. More specifically, since the core of evaluation metrics is to compare different recommendation models' performance, they observed that the relative order of models' performance is not maintained when utilizing sampled metrics in comparison to the original global metrics. For example, Model A has a larger value than Model B in terms of $Recall@10$ globally ($Recall_A@10 > Recall_B@10$), whereas it would be the opposite order when using a sampled metric ($Recall_A@10 < Recall_B@10$). They claim that a "sampled metric can be a poor indicator of the true performance of recommender algorithms" [24]. Thus, cautionary use is suggested or even sampling should be avoided for metric calculation.

This claim challenges a considerable amount of work [10, 16, 17, 23, 36–38] within the recommendation community. What are the implications for existing studies that utilize sampling top-$K$ criteria? Does this render their results somewhat invalid? Does it imply that a meaningful top-$K$ evaluation requires the use of all items? To be able to firmly answer these questions, a better understanding of the sampling-based top-$K$ metrics is much needed. In the meantime, a sampling approach, where acceptable, can be a useful tool for saving computational costs and speeding up evaluation time. Although computational resources might not be a big problem for enormous mega-corporations, such as Google or Amazon, for many smaller, resource-constrained organizations and businesses, it may still be an issue. For instance, if valid, a sampling approach can be a quick way to help evaluate the promise of a given algorithm, screening for the eventual exact/global top-$K$ evaluation.

***Contribution 1: A Sampling-Based Top-K Recall Metric Can Be Mapped to the Global One at f(K).*** Krichene and Rendle [24] state that a sampling-based top-$K$ metric cannot properly reflect the global metric at the same $K$. Slightly contrary to the claim, we propose that there exists an approximately linear mapping function $f$ such that the $Recall@K$ metric in sampling-based evaluation is approximate to the $Recall@f(K)$ in global evaluation, where $K$ and $f(K)$ represent different top-$K$ selections. We take Figure 1 as an example to intuitively explain the insight. The left side of Figure 1 plots the top-$K$ global Recall curve, where each point $(K, Recall@K)$ on the curve is a top-$K$ metric. Here the range of $K$ is from 1 to $N$ (total number of items). The middle part of Figure 1 is the top-$K$ sample Recall curve. Since the rank $r_u$ is obtained in the sample set ($n = 100$), the range of sampling-based top-$K$ Recall is from 1 to $n = 100$. As we discovered, there is an approximately linear relation between global and sample metrics. One could come up with some mapping functions that align the sample Recall curve to the global scale (shown on the right side of Figure 1). These findings we make could partially save the amount of sampling-based evaluation work [10, 17, 23, 36–38] in danger.

***Efforts to Estimate Global Metrics.*** Since the sampled metrics are inconsistent with the global metrics [24], to make the sampled metric useful, at the same time, Krichene and Rendle [24] heuristically proposed a few estimators to correct the sampled metrics. The first correction that they used is an unbiased estimator of the rank. They tried to find vector-liked corrected metrics $\hat{\mathcal{F}}$ that minimizes the following equation: $\arg\min_{\hat{\mathcal{F}} \in \mathbb{R}^n} \sum_{R=1}^{N} p(R)(\mathbb{E}_r[\hat{\mathcal{F}}_r|R] - \mathcal{F}(R))^2$, where $n$ is the sample set size, $N$ is the total number of items, $\mathcal{F}(\cdot)$ is the ground truth global metric function, $\hat{\mathcal{F}}(\cdot)$ is the corrected metric function applied to sampled rank, and $p(R)$ is a prior on the distribution of ranks. The central idea of this equation is trying to come up with a correction of the metric function $\hat{\mathcal{F}}(\cdot)$ so that once it is applied to sampled rank $\{r_u\}_{u=1}$, one can still derive similar results as the original metric $\mathcal{F}(\cdot)$ applied to global rank $\{R_u\}_{u=1}$. As pointed out by Krichene and Rendle [24], the potential issue with this estimator is that it could raise high variance practically. Thus, they propose

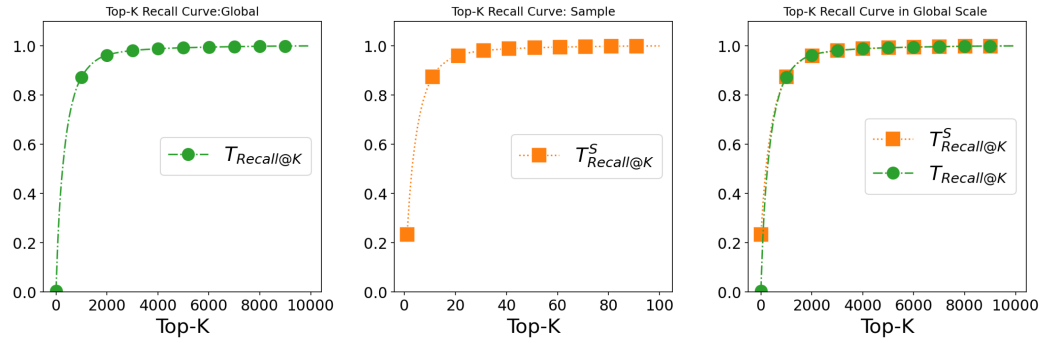


Fig. 1. Curve relationship of model NeuMF on the *pinterest-20* dataset. Left: $T_{Recall@K}$ is the top-$K$ global Recall curve. Middle: $T^S_{Recall@K}$ is the sampling top-$K$ Recall curve. Right: $T^S_{Recall@K}$ curve mapped to the global scale by a baseline mapping function.

another BV estimator by introducing a variance term, which borrows an idea from **Bias-Variance (BV)** tradeoff: $\arg\min_{\widehat{\mathcal{F}} \in \mathbb{R}^n} \sum_{R=1}^{N} p(R)((\mathbb{E}_r[\widehat{\mathcal{F}}_r|R] - M(R))^2 + \gamma \cdot Var_r[\widehat{\mathcal{F}}_r|R])$, where $\gamma$ is a positive constant.

***Contribution 2: Solution for Global Metrics Estimation.*** The preceding BV estimator [24] does not directly minimize the expected errors between the item-sampling-based top-$K$ metrics and the global top-$K$ metrics. In addition, there are no optimality results that have been established for that expected error. To tackle this issue, we propose two classes of methods. Method 1 is *estimating global ranking distribution*. We bring in a new intermediate fundamental problem—estimate the global rank distribution $\{R_u\}_{u=1}^M$ given the sampling-based rank information $\{r_u\}_{u=1}^M$, where $M$ is the total number of user and $R_u$ is the global rank of target item for user $u$ and $r_u$ is that of sampling-based rank. By theoretically analyzing the sampling process, we derive the statistical relations between these two rank distributions. We leverage MLE (Maximum Likelihood Estimation) and MES (Maximal Entropy with Squared distribution distance) methods to help estimate the global rank distribution. As long as the global rank distribution is obtained, the expectation of global metrics can be further inferred. Although these two methods are comparable to the BV estimator in the work of Krichene and Rendle [24], a more accurate estimator is still needed. Method 2 is *multinomial global metric estimation*. Similar to Krichene and Rendle [24], where global metrics are directly estimated, we propose to find the expected estimation of the global metric by directly optimizing the expectation error between the corrected metrics and the original one: $\mathbb{E}[\frac{1}{M}\sum_{u=1}^{M}\widehat{\mathcal{F}}(r_u) - \sum_{R=1}^{N} P(R)\cdot\mathcal{F}(R)]^2$. We then highlight subtle differences from the BV estimator derived and point out the potential issues of the BV estimator because it fails to link the user population size with the estimation variance.

***Contribution 3: Adaptive Sampling Could Improve Estimating Accuracy and Solve the "Blind Spot" Issue.*** Despite the preceding efforts, we still face the "blind spot" issue, where the estimation can be quite inaccurate when $K$ is small. In offline evaluation, we are interested in the top-ranked items and top-$K$ metrics, when $K$ is relatively small. However, the current item-sampling estimation seems to have a "blind spot" for the top-rank distribution. For example, when there are $n = 100$ samples and $N = 10k$, the estimation granularity is only at around the 1% ($1/n$) level [24, 26]. We can only infer that the top items in the samples are the top 1% (top 100) in the global rank, and we are unable to further tell whether the top items in the sample set are in the top-50, for example. Given this, even with the best estimator for the item sampling, we may still not be able to provide accurate results for the top-$K$ metrics. A remedy is increasing the sampling

Table 1. Notations for Leave-One-Out Evaluation Setting

| | |
|---|---|
| $U$ | The set of overall users, and $\|U\| = M$ |
| $I$ | The set of overall items, and $\|I\| = N$ |
| $M$ | Total # of users |
| $N$ | Total # of items |
| $i_u$ | The target item (to be ranked) for each user $u$ |
| $I_u$ | Sampled test set for user $u$, consisting of 1 target item $i_u$, $n-1$ sampled items |
| $n$ | Sample set size, $\|I_u\| = n$ |
| $R_u$ | Rank of item $i_u$ among all items $I$ for user $u$ |
| $r_u$ | Rank of item $i_u$ among $I_u$ for user $u$ |
| $T$ | Evaluation metric (for a recommendation model) (e.g., $Recall$, $Recall@K$) |
| $\mathcal{F}$ | Individual evaluation metric function (for each item rank) (e.g., $Recall(\cdot)$, $Recall@K(\cdot)$) |
| $T^S$ | Sampled evaluation metric |
| $\widehat{T}$ | Estimated evaluation metric |

size, but it can significantly increase the estimation cost too, limiting the benefits of item sampling. To solve this issue, we introduce the adaptive sampling method. Intuitively, instead of sampling a fixed number of negative items during evaluation, we dynamically sample negative items for different users, leading to an informative sampling-based rank distribution. By leveraging MLE and the **Expectation-Maximization (EM)** algorithm, we are able to derive an adaptive estimator that exhibits efficiency and effectiveness.

***Contribution 4: The User Sampling Problem.*** In addition, we bring in another interesting sampling-based evaluation problem where the number of users is much more than that of the item. In this case, sampling from the user perspective and deriving the estimated metrics could lead to more efficient and accurate results.

In summary, in this article, we thoroughly investigate the item-sampling-based recommendation evaluation problem. We build a connection between the sampled top-$K$ Recall metric and its global counterpart. To resolve the inconsistent issue, we propose several estimators to help estimate accurate global metrics. Then we stress the "blind spot" issue that was overlooked and propose an adaptive method to tackle this task. Finally, we take another perspective to investigate the user sampling approach. This extensive and thorough study helps build a complete theoretical foundation for sampling-based recommendation evaluation problems. It also provides evidence for making item sampling a useful tool for recommendation evaluation.

The rest of the article is structured as follows. Section 2 introduces the background and preliminaries, and provides an overview of previous efforts of metric estimation. Section 3 (Contribution 1) reveals the linear relation between sampling-based Recall and the global one. Section 4 (Contribution 2; Method 1) discusses various sampling-based estimators by estimating global rank distribution. Section 5 (Contribution 2; Method 2) presents estimators directly from optimizing the expectation error. Section 6 (Contribution 3) proposes the novel adaptive sampling and estimation method. Section 7 (Contribution 4) explores sampling evaluation from the user perspective. Section 8 reports on the experimental results. Finally, Section 9 offers concluding remarks.

## 2 BACKGROUND

### 2.1 Leave-One-Out Recommendation Evaluation Setting

There are a user set $U$, ($|U| = M$) and a item set $I$, ($|I| = N$) Table 1. Assume each user $u$ is associated with one and only one target item $i_u$ (hold out from the training set for $u$). A recommendation

model $A$ is trained on the training set and would compute a personalized rank $R_u = A(i_u|u, I)$ for item $i_u$ among all the items $I$, and $\{R_u\}_{u=1}^M$ is called *global rank* distribution. In contrast, one can also compute the other type personalized rank $r_u = A(i_u|u, I_u)$ for item $i_u$ among sampled set $I_u = \{ i \sim I \backslash i_u \} \cup \{i_u\}$, $|I_u| = n$. In this case, $\{r_u\}_{u=1}^M$ is called *sampled rank* distribution. Given $\{R_u\}_{u=1}^M$ or $\{r_u\}_{u=1}^M$, the (global or sampling-based) evaluation metrics can be computed according to specific metric functions.

### 2.2 Global Top-*K* Evaluation Metrics

A metric function $\mathcal{F}$ maps its integer input (any rank $R_u$) to a real-valued score. The aggregation of these scores over all the users is called *(global) metric* $T$:

$$T = \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}(R_u). \tag{1}$$

It is worth noting that a metric function $\mathcal{F}$ is an operation on some integer rank input and a metric $T$ is a corresponding aggregation of the output, which is a real value and represents the performance of a recommendation model. When we talk about *Recall*, *NDCG* or *Recall@K*, and *NDCG@K*, they can be both metric functions and metrics, depending on the context. Similar to Krichene and Rendle [24], we define the simplified top-$K$ metric functions $\mathcal{F}$ in the following ways:

$$\mathcal{F}_{Recall@K}(x) = \delta(x \le K), \quad \mathcal{F}_{NDCG@K}(x) = \delta(x \le K) \cdot \frac{1}{\log_2(x+1)}, \quad \mathcal{F}_{AP@K}(x) = \delta(x \le K) \cdot \frac{1}{x}, \tag{2}$$

where $\delta(x) = 1$ if $x$ is true and 0 otherwise. The corresponding global metrics $T$ for a given distribution $\{R_u\}_{u=1}^M$ are

$$\begin{aligned} T_{Recall@K} &= \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}_{Recall@K}(R_u) = \frac{1}{M} \sum_{u=1}^{M} \delta(R_u \le K) \\ T_{NDCG@K} &= \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}_{NDCG@K}(R_u) = \frac{1}{M} \sum_{u=1}^{M} \delta(R_u \le K) \cdot \frac{1}{\log_2(R_u+1)} \\ T_{AP@K} &= \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}_{AP@K}(R_u) = \frac{1}{M} \sum_{u=1}^{M} \delta(R_u \le K) \cdot \frac{1}{R_u}. \end{aligned} \tag{3}$$

### 2.3 Sampling-Based Top-*K* Evaluation Metrics

As aforementioned in Sections 1 and 2.1, it is also a common choice [5, 10, 16, 17, 22, 23, 36–38] to use sampling-based top-$K$ metrics to evaluate recommendation models, denoted as $T^S$ in general:

$$T^S = \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}(r_u). \tag{4}$$

It is obvious that $r_u$ and $R_u$ differ substantially—for example, $r_u \in [1, n]$, whereas $R_u \in [1, N]$. Therefore, for the same $K$, the item-sampling-based top-$K$ metric $T^S$ and the global top-$K$ metric $T$ correspond to distinct measures (no direct relationship): $T \neq T^S$ ($T_{Recall@K} \neq T^S_{Recall@K}$ even in expectation). This problem is highlighted in other works [24, 32], referring to these two metrics being *inconsistent*. From the perspective of statistical inference [25], the basic sampling-based top-$K$ metric $T^S$ is not a *reasonable* or good *estimator* of $T$.

### 2.4 Statistical View of the Sampling Process: Sampling with Replacement (Binomial Distribution)

For a given user $u$, let $X_u$ denote the number of sampled items that are ranked in front of relevant item $i_u$:

$$X_u = \sum_{i=1}^{n-1} X_{ui}, \quad X_{ui} \sim Bernoulli\left(b_u = \frac{R_u - 1}{N - 1}\right),$$

where $X_{ui}$ is a Bernoulli random variable for each sampled item $i$: $X_{ui} = 1$ if item $i$ has rank range in $[1, R_u - 1]$ ($b_u$ is the corresponding probability) and $X_{ui} = 0$ if $i$ is located in $[R_u + 1, N]$. Thus, $X_u$ follows binomial distribution:

$$X_u \sim Binomial\left(n - 1, b_u = \frac{R_u - 1}{N - 1}\right). \tag{5}$$

And the random variable $r_u = X_u + 1$, and we have

$$p_u = CDF(K; n - 1, b_u) = Pr(r_u \le K) = \begin{cases} \sum_{l=0}^{k-1} \binom{n-1}{l} b_u^l (1 - b_u)^{n-1-l} & , R_u \ge K \\ 1 & , R_u < K. \end{cases}$$

### 2.5 Efforts Toward Metric Estimation

Given the sampling ranked results in the test dataset, $\{r_u\}_{u=1}^M$, how to infer/approximate the $T$ (Equation (1)) without the knowledge $\{R_u\}_{u=1}^M$? The work of Krichene and Rendle [24] is the most closely related work that studies metric estimation problems.

***Krichene and Rendle's Approaches.*** Krichene and Rendle [24]develop a discrete corrected metric function $\widehat{\mathcal{F}}(r)$ to approach:

$$T = \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}(R_u) \approx \frac{1}{M} \sum_{u=1}^{M} \widehat{\mathcal{F}}(r_u) = \sum_{r=1}^{n} \tilde{P}(r) \widehat{\mathcal{F}}(r) = \widehat{T}, \tag{6}$$

where $\tilde{P}(r)$ is the empirical rank distribution on the sampling data. They have proposed a few estimators based on this idea, including estimators that use unbiased rank estimators, with monotonicity constraint (*CLS*):

$$\arg\min_{\hat{\mathcal{F}} \in \mathbb{R}^n} \sum_{R=1}^{N} p(R) (\mathbb{E}_r[\widehat{\mathcal{F}}_r | R] - \mathcal{F}(R))^2 \tag{7}$$

and utilize BV tradeoff:

$$\arg\min_{\widehat{\mathcal{F}} \in \mathbb{R}^n} \sum_{R=1}^{N} p(R) \left( (\mathbb{E}_r[\widehat{\mathcal{F}}_r | R] - M(R))^2 + \gamma \cdot Var_r[\widehat{\mathcal{F}}_r | R] \right)^2 \tag{8}$$

Their study shows that only BV is competitive [24] and the solution is

$$\widehat{\mathcal{F}} = \left( (1.0 - \gamma) A^T A + \gamma \mathrm{diag}(\boldsymbol{c}) \right)^{-1} A^T \boldsymbol{b}, \tag{9}$$

$$A \in \mathbb{R}^{N \times n}, \quad A_{R,r} = \sqrt{P(R)} P(r|R), \quad \boldsymbol{b} \in \mathbb{R}^N, \quad b_R = \sqrt{P(R)} \mathcal{F}(R), \quad \boldsymbol{c} \in \mathbb{R}^n, \quad c_r = \sum_{R}^{N} P(R) P(r|R). \tag{10}$$

## 3 TOP-*K* RECALL METRIC ESTIMATION VIA MAPPING FUNCTION

In this section, we would like to introduce our first finding—the linear mapping relation between the sampled top-*K* *Recall* metric and the global one. In short, there exists a function $f(K)$ such that the sampled top-*K* *Recall* metric is approximately equal to the global top-$f(K)$ *Recall* metric: $T^S_{Recall@K} \approx T_{Recall@f(K)}$.

### 3.1 Statistical View of the *Recall* Metric

Let us consider the definition of the global top-*K* *Recall* (or Hit-Ratio) metric (rewrite Equation (3)):

$$T_{Recall@K} = \frac{1}{M}\sum_{u=1}^{M}\delta(R_u \le K) = \sum_{R=1}^{N}\tilde{P}(R)\cdot\delta(R \le K), \tag{11}$$

where $\tilde{P}(R)$ is the frequency of users with item $i_u$ rank in position $R$, also denoted as *empirical global rank distribution*:

$$\tilde{P}(R) = \frac{1}{M}\sum_{u=1}^{M}\delta(R_u = R). \tag{12}$$

Next, let us revisit the top-*K* Recall (Hit-Ratio) under-sampling in Equation (4). For a given user $u$ and the relevant item $i_u$, we first sample $n-1$ items from the entire set of items $I$, forming the subset $I_u$ (including $i_u$). Let the relative rank of $i_u$ among $I_n$ be denoted as $r_u = A(i_u|u, I_u)$. Note that $r_u$ is a random variable depending on the sampling set $I_u$.

Given this, the sampling top-*K* Recall metric can be written as follows:

$$T^S_{Recall@K} = \frac{1}{M}\sum_{u=1}^{M}Z_u, \quad Z_u \sim Bernoulli(p_u = Pr(r_u \le K)), \tag{13}$$

where $Z_u$ is a random variable for each user $u$ and follows a Bernoulli distribution with probability $p_u = Pr(r_u \le k)$, where $p_u$ is defined in Section 2.4. Now, recall that we are trying to study the relation between $T^S_{Recall@K}$ and $T_{Recall@K}$. We note that the population sum $\sum_{u=1}^{M} Z_u$ is a Poisson binomial distributed variable (a sum of $M$ independent Bernoulli distributed variables). Its mean and variance will simply be sums of the mean and variance of the $n$ Bernoulli distributions:

$$\mu = \sum_{u=1}^{M}p_u, \quad \sigma^2 = \sum_{u=1}^{M}p_u(1-p_u).$$

Given this, the expectation and variance of $T^S_{Recall@K}$ in Equation (13):

$$\mathbb{E}[T^S_{Recall@K}] = \frac{1}{M}\sum_{u=1}^{M}p_u = \sum_{R=1}^{N}\tilde{P}(R)\cdot P(R), \tag{14}$$

$$Var[T^S_{Recall@K}] = \frac{1}{M^2}\sum_{u=1}^{M}p_u(1-p_u) = \frac{1}{M}\sum_{R=1}^{N}\tilde{P}(R)\cdot P(R)(1-P(R)), \tag{15}$$

where $P(R)$ is the probability that users who are in the same group ($R_u = R$) share the same $p_u$.

### 3.2 A Functional View of $T_{Recall@K}$ and $T^S_{Recall@K}$

To better understand the relationship between $T_{Recall@K}$ (global top-*K* Recall) and $T^S_{Recall@K}$ (the sampling version), it is beneficial to take a functional view of them. Let $\mathcal{R}$ be the random variable

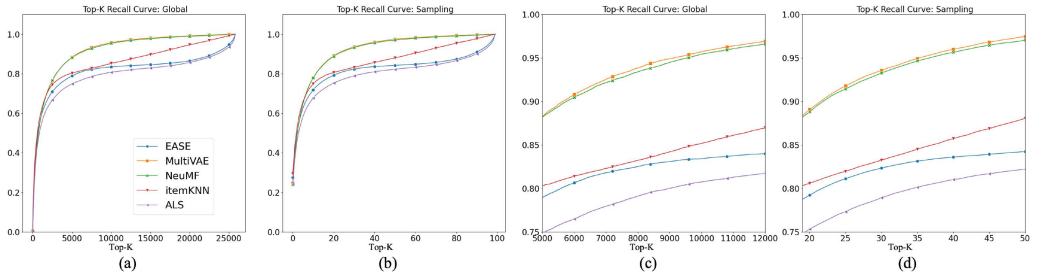


Fig. 2. Global vs sampling top-$K$ Hit-Ratio on the *yelp* dataset. To display the details clearly, we zoom in on the global Recall curves (a) and the sampling Recall curves (b) at different range scales to (c) and (d), respectively. Comparing the two figures, we can easily conclude that sampling evaluation maintains the same curve trend as global evaluation for different algorithms even at small error range.

for the user's item rank, with probability mass function $Pr(\mathcal{R} = R)$; then, $T_{Recall@K}$ is simply the empirical cumulative distribution of $\mathcal{R}$ ($\widehat{Pr}$):

$$T_{Recall@K} = \widehat{Pr}(\mathcal{R} \leq K), \quad \tilde{P}(R) = \widehat{Pr}(\mathcal{R} = R). \tag{16}$$

For $T^S_{Recall@K}$, its direct meaning is more involved and will be examined in the following. For now, we note that $T^S_{Recall@K}$ is a function of $K$ varying from 1 to $n$, where $n-1$ is the number of sampled items.

Figure 2(a) displays the curves of empirical accumulative distribution $T_{Recall@K}$ (a.k.a. the global top-$K$ Hit-Ratio, varying $K$ from 1 to $N = 25{,}815$), for five representative recommendation algorithms (three classical and two deep learning methods), on the *yelp* dataset. To observe the performance of these methods more closely, we zoom in on the range $K$ from 5,000 to 12,000 in Figure 2(c). Figure 2(b) displays the curves of functional fitting of function $T^S_{Recall@K}$ (the sampling top-$K$ Hit-Ratio, varying $K$ from 1 to $n = 100$) with $n-1$ samples, under sampling with replacement, for the same five representative recommendation algorithms on the same dataset. Similarly, we zoom in and highlight $K$ from 20 to 50 in Figure 2(d).

How can the sampling Recall curves help to reflect what happened in the global curves? Before we consider the more detailed relationship between them, we introduce the following results:

Theorem 3.1 (Sampling Theorem). *Let us assume we have two global* Recall curves *(empirical cumulative distribution),* $T^{(1)}_{Recall@K}$ *and* $T^{(2)}_{Recall@K}$*, and assume one curve dominates the other one, (i.e.,* $T^{(1)}_{Recall@K} \geq T^{(2)}_{Recall@K}$ *for any* $1 \leq K \leq N$*); then, for their corresponding sampling curve at any* $k$ *for any size of sampling, we have*

$$\mathbb{E}[T^{S,(1)}_{Recall@K}] \geq \mathbb{E}[T^{S,(2)}_{Recall@K}].$$

The preceding theorem shows that, under the strict order of global Recall curves (although it may be quite applicable for searching/evaluating better recommendation algorithms, like in Figure 2), sampling Hit-Ratio curves can maintain such order. However, this theorem does not explain the stunning similarity, shapes, and trends shared by the global and their corresponding sampling curves. Basically, the detailed performance differences among different recommendation algorithms seem to be well preserved through sampling. However, unless $n \approx N$, $T^S_{Recall@K}$ does not correspond to $T_{Recall@K}$ (as in what is being studied by Rendle [32]). Those observations hold on other datasets and recommendation algorithms as well, not only on this dataset. Thus, intuitively and through the preceding experiments, we may conjecture that it is the overall curve

$T_{Recall@K}$ that is being approximated by $T^S_{Recall@K}$. Since these functions are defined on different domain sizes $N$ vs $n$, we need to define such approximation carefully and rigorously.

### 3.3 Mapping Function $f$

To explain the similarity between the global and sampling top-$K$ Recall curves, we hypothesize the following:

> There exists a function $f(K)$ such that the relation $T^S_{Recall@K} \approx T_{Recall@f(K)}$ holds for different ranking algorithms on the same dataset.

In a way, the sampling metric $T^S_{Recall@K}$ is like "signal sampling" [31], where the global metrics between top 1 to $N$ are sampled (and approximated) at only $f(1) < f(2) < \cdots < f(n)$ locations, which corresponds to $T^S_{Recall@K}$ $(k = 1, 2, \ldots, n)$. In general, $f(k) \neq k$ (when $n << N$) ( [32]).

To identify such a mapping function, let us take a look at the error between $T^S_{Recall@K}$ and $T_{Recall@f(K)}$:

$$|T^S_{Recall@K} - T_{Recall@f(K)}| \leq |T^S_{Recall@K} - \mathbb{E}[T^S_{Recall@K}]| + |\mathbb{E}[T^S_{Recall@K}] - T_{Recall@f(K)}|. \quad (17)$$

Thanks to the Hoeffding's bound, we observe

$$Pr(|T^S_{Recall@K} - \mathbb{E}[T^S_{Recall@K}]| \geq t) \leq 2\exp(-2Mt^2).$$

This can be a rather tight bound, due to the large number of users in the population. For example, if $M = 30K, t = 0.01$,

$$Pr(|T^S_{Recall@K} - \mathbb{E}[T^S_{Recall@K}] \geq 0.01| \leq 0.005.$$

If we want to look more closely, we may use the law of large numbers and utilize the variance in Equation (15) for deducing the difference between $T^S_{Recall@K}$ and its expectation. Overall, for a large user population, the sampling top-$k$ Hit-Ratio will be tightly centered around its mean. Furthermore, if the user number is, indeed, small, an average of multiple sampling results can reduce the variance and error. In the publicly available datasets, we found that one set of samples is typically very close to the average of multiple runs.

Given this, our problem is *how to find the mapping function $f$ such that $|\mathbb{E}[T^S_{Recall@K}] - T_{Recall@f(K)}|$ can be minimized (ideally close to or equal to 0)*. Note that $f$ should work for all $K$ (from 1 to $n$), and it should be independent of algorithms on the same dataset.

### 3.4 Approximating Mapping Function $f$

***Baseline.*** To start, we may consider the following naive mapping function. We notice that for any $n$,

$$\mathbb{E}[X_u] = (n-1) \cdot b_u = (n-1)\frac{R_u - 1}{N-1} = \mathbb{E}[r_u] - 1, \quad \mathbb{E}[r_u] = 1 + (n-1)\frac{R_u - 1}{N-1}.$$

When $n$ is large, we simply use the indicator function $\delta(\mathbb{E}[r_u]) \leq K$ to approximate and replace $Pr(r_u \leq K)$. Thus,

$$1 + (n-1)\frac{R_u - 1}{N-1} \leq K, \quad R_u \leq \frac{K-1}{n-1} * (N-1) + 1 \triangleq f(K). \quad (18)$$

To wrap to, this baseline function $f(K)$ enable us that: for any given $T^S_{Recall@K}$, we can obtain its approximation - $T_{Recall@K'}$, where $K' = f(K)$. This guarantees us that we can plot the $T^S_{Recall@K}$ curve in the global range (1 to $N$) as well, which would help us directly observe the relation between $T^S_{Recall@K}$ and $T_{Recall@K}$ (Figure 3).

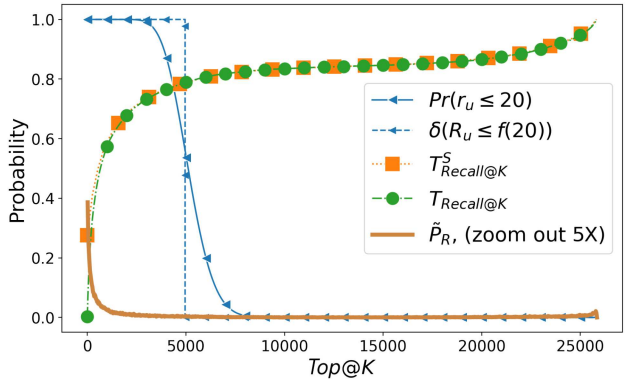


Fig. 3. Curve relationship of the model EASE on the *yelp* dataset. $T_{Recall@K}$ is the top-$K$ global Recall curve; $T^S_{Recall@K}$ is the sampling top-$K$ Recall curve shown in global scale (by baseline Equation (18)); and $\tilde{P}_R$ is the empirical user ranking distribution, where we make it five times larger (multiply) for displaying purpose.

Since the indicator function, $\delta(\mathbb{E}[r_u] \leq K)$ is a rather crude estimation of the CDF of $r_u$ at $K$, and this only serves as a baseline for our approximation of the mapping function $f$.

***Approximation Requirements.*** Before we introduce more carefully designed approximations of the mapping function $f$, let us take a close look at the expectation of the sampling top-$K$ Recall $\mathbb{E}[T^S_{Recall@K}]$ and $T^S_{Recall@f(K)}$. Figure 3 shows how the user empirical probability mass function $\tilde{P}(R)$ works with the step indicator function $\delta(R_u \leq f(K))$, and $b_u$ (assuming a hypergeometric distribution), to generate the global top-$K$ and sampling Recall.

We have the following requirements:

- *Existence of mapping function $f$ for each individual $T_{Recall@K}$ curve*: Given any $K$, assuming $T_{Recall@f(K)}$ is a continuous cumulative distribution function (i.e., assuming that there is no jump/discontinuity on the CDF, and that $f(K)$ is a real value), then there is $f(K)$ such that $T_{Recall@f(K)} = \mathbb{E}[T^S_{Recall@K}]$. In our problem setting, where $f(K)$ is integer valued and ranges between 0 and $N$, the best $f(K)$, theoretically, is

$$f(K) = \arg_f \min |T_{Recall@f(K)} - \mathbb{E}[T^S_{Recall@K}]|.$$

- *Mapping function $f$ for different $T_{Recall@f(K)}$ curves*: Since our main purpose is for $T^S_{Recall@K}$ to be comparable across different recommendation algorithms, we prefer $f(K)$ to be the same for different Recall curves (on the same dataset). Thus, by comparing different $T^S_{Recall@K}$, we can infer their corresponding Recall $T_{Recall}$ at the same $f(K)$ location. Figure 2 and Figure 3 show that the sampling Recall curves are comparable with respect to their respective counterparts and suggest that such a mapping function, indeed, may exist.

But how does the second requirement coexist with the first requirement of the minimal error of individual curves? We note that for most of the recommendation algorithms, their overall *Recall* curve $T_{Recall@K}$ is actually fairly similar (see Figure 2). From another viewpoint, if we allow individual curves to have different optimal $f(K)$, the difference (or shift) between them is rather small and does not affect the performance comparison between them, using the sampling curves $T^S_{Recall@K}$. In this section, we will focus on studying dataset-algorithm-independent mapping functions.

### 3.5 Boundary Condition Approximation

Consider that sampling with replacement, for any individual user, $X_u$ from Equation (5), obeys binomial distribution. Apply the general case of bounded variables Hoeffding's inequality:

$$Pr(|X_u - \mathbb{E}[X_u]| \geq t) \leq 2e^{-\frac{2t^2}{n-1}}$$

since $r_u = X_u + 1$, and $\mathbb{E}[r_u] = \mathbb{E}[X_u] + 1 = (n-1)b_u + 1$:

$$\begin{cases} Pr(r_u \geq (n-1)b_u + 1 + t) \leq 2e^{-\frac{2t^2}{n-1}} \\ Pr(r_u \leq (n-1)b_u + 1 - t) \leq 2e^{-\frac{2t^2}{n-1}}. \end{cases} \tag{19}$$

The preceding inequalities indicate that $r_u$ is restricted around its expectation within the range defined by $t$. The second term of error in Equation (17) can be written as follows:

$$\mathbb{E}[T^S_{Recall@K}] - T_{Recall@f(K)} = -\sum_{R=1}^{f(K)} \tilde{P}(R) \cdot Pr(r_R \geq k+1) + \sum_{R=f(K)+1}^{N} \tilde{P}(R) \cdot Pr(r_R \leq K), \tag{20}$$

where $r_R = r_u$ for $R_u = R$. For some relatively large $t$ (compared to $\sqrt{n-1}$), the probability in Equation (19) can come extremely close to 0. Based on this fact, if we would like to limit the first term $Pr(r_R \leq K+1)$ to approach 0, $K+1$ must be greater than $(n-1)b_u + 1 + t$. Similar to the second term, we have

$$\begin{cases} r_u \geq K+1 \geq (n-1)\frac{R-1}{N-1} + 1 + t), & R = 1, \ldots, f^{lower}(K) \\ r_u \leq K \leq (n-1)\frac{R-1}{N-1} + 1 - t, & R = f^{upper}(k)+1, \ldots, N, \end{cases}$$

where $f^{lower}(K)$ and $f^{upper}(K)$ are the lower bound and upper bound for $f(K)$, respectively. Explicitly,

$$f^{lower}(K) \leq (K-t) \cdot \frac{N-1}{n-1} + 1, \quad f^{upper}(K) \geq (K+t-1) \cdot \frac{N-1}{n-1}. \tag{21}$$

Given this, define $f$ as the average of above two bounds

$$\boxed{f(K) = \left\lfloor \frac{f^{lower}(K) + f^{upper}(K)}{2} \right\rfloor = \left\lfloor \left(K - \frac{1}{2}\right) \frac{N-1}{n-1} + \frac{1}{2} \right\rfloor}. \tag{22}$$

Note that although this formula appears similar to our baseline Equation (18), the difference between them is actually pretty big ($\approx \frac{1}{2}\frac{N-1}{n-1}$). As we will show in the experimental results, this formula is remarkably effective in reducing the error $|\mathbb{E}[T^S_{Recall@K}] - T_{Recall@f(K)}|$.

### 3.6 Beta Distribution Approximation

In this approach, we try to directly minimize $\mathbb{E}[T^S_{Recall@K}] - T_{Recall@f(K)}$, and this is equivalent to

$$\sum_{R=1}^{N} \tilde{P}(R) \cdot \delta(R \leq f(K)) = \sum_{R=1}^{N} \tilde{P}(R) \cdot Pr(r_R \leq K). \tag{23}$$

To get a closed-form solution of $f(K)$ from the preceding equation, we leverage the Beta distribution $Beta(a, 1)$ to represent the user ranking distribution $\tilde{P}(R)$, inspired by Li et al. [28]: $\tilde{P}(R) = \frac{1}{\mathcal{B}(a,1)}(\frac{R-1}{N-1})^{a-1}\frac{1}{N-1}$, where $a$ is a constant parameter and $\frac{1}{N-1}$ is the constant for discretized Beta distribution. Note that $\frac{R-1}{N-1}$ normalizes the user rank $R_u$ from $[1, N]$ to $[0, 1]$. Especially when

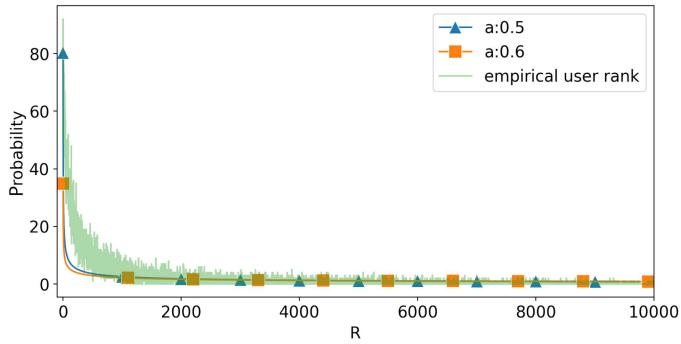


Fig. 4. Beta distributions and empirical user rank distribution $\tilde{P}(R)$.

$a < 1$, this distribution can represent exponential distribution, which can help provide fit for the *Recall* distribution. Figure 4 illustrates the Beta distribution fitting of $\tilde{P}(R)$.

According to the detailed derivation in Appendix B, we have the following recurrent formula:

$$f(k+1;a) = \left[a[N-1]^a \binom{n-1}{k} \mathcal{B}(a+k, n-k)[f(k;a)-1]^a\right]^{1/a} + 1, \tag{24}$$

$$f(1;a) = (N-1)[a\mathcal{B}(a,n)]^{1/a} + 1. \tag{25}$$

Figure 5 shows the relative difference of all $f(k;a)$ sequences. (For detailed analysis please refer to Appendix C).

## 4 TOP-*K* METRIC ESTIMATION VIA GLOBAL RANK DISTRIBUTION LEARNING

### 4.1 Learning the Empirical Rank Distribution Problem

In this section, our new proposed approach is based on the following observation:

$$T = \frac{1}{M}\sum_{u=1}^{M} \mathcal{F}(R_u) = \sum_{R=1}^{K} \tilde{P}(R) \cdot \mathcal{F}(R). \tag{26}$$

Again, $T$ is any metric to quantify the quality of a recommendation model and $\mathcal{F}$ is the corresponding specific metric function in Section 2.2. Thus, if we can estimate $\widehat{P}(R) \approx P(R) \approx \tilde{P}(R)$, then we can derive any metric estimator as

$$\widehat{T} = \sum_{R=1}^{K} \widehat{P}(R) \cdot \mathcal{F}(R). \tag{27}$$

Given this, we introduce the new problem of learning and estimating the empirical rank distribution $\{\tilde{P}(R)\}_{R=1}^{N}$ based on sampling $\{r_u\}_{r=1}^{M}$. To our knowledge, this problem has not been formally and explicitly studied before for sampling-based recommendation evaluation.

The importance of the problem is twofold: on one side, the learned empirical rank distributions can directly provide estimators for any metric $T$; on the other side, since this question is closely related to the underlying mechanism of sampling for recommendation, tackling it can help better understand the power of sampling and resolve the questions as to if and how we should use sampling for evaluating recommendation. Furthermore, since metric $T$ is the linear function of $\{\tilde{P}(R)\}_{R=1}^{K}$, the statistical properties of estimator $\widehat{P}(R)$ can be nicely preserved by $\widehat{T}$ [25]. In addition, this approach can be considered as metric independent: We only need to estimate the empirical rank distribution $\tilde{P}(R)$ once; then we can utilize it for estimating all the top-$K$ evaluation metrics $T$ (including for different $K$) based on Equation (27).

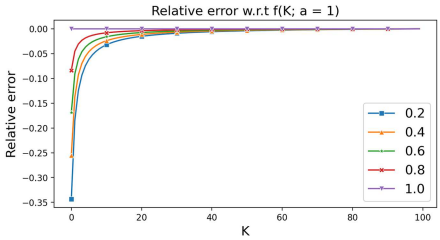


Fig. 5. Relative error w.r.t. $f(k; a = 1)$. Example on the *yelp* dataset, $n = 100$.

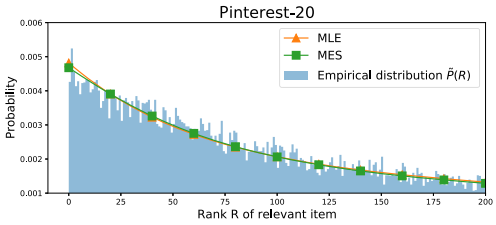


Fig. 6. Learning empirical rank distribution $P(R)$.

For the first time, we show that the BV estimator [24] can also be used to estimate $P(R)$:

$$\widehat{P}(R) = \widehat{\mathcal{F}}_{BV}(R) - \widehat{\mathcal{F}}_{BV}(R-1) = (\tilde{P}(r))_{r=1}^{n}\Big((1.0-\gamma)A^TA + \gamma \text{diag}(\boldsymbol{c})\Big)^{-1}A^T \cdot \boldsymbol{b}_R, \tag{28}$$

where $\widehat{\mathcal{F}}_{BV}(R)$ is the BV estimator for the *Recall@R* metric function, $(\tilde{P}(r))_{r=1}^{n}$ is the row vector of empirical rank distribution over the sampling data, and $\boldsymbol{b}_R$ has the $R$-th element as $b_R$ (Equation (10)) and other elements as 0. We consider this as our baseline for learning the empirical rank distribution.

In the following, we introduce a list of estimators for the empirical rank distribution $\{P(R)\}_{R=1}^{N}$ based on sampling ranked data: $\{r_u\}_{r=1}^{M}$. Figure 6 sketches the different approaches to learning the empirical rank distribution $P(R)$, including the MLE and the maximal entropy based approach (MES).

### 4.2 Sampling Rank Distribution: Mixtures of Binomial Distributions

Assume an item $i$ is ranked $R$ in the entire set of items $I$. Then there are $R-1$ items whose rank is higher than item $i$. Under the (uniform) sampling (with replacement), the probability of picking up an item with a higher rank than $R$ is $\theta := \frac{R-1}{N-1}$. Let $x$ be the number of irrelevant items ranked in front of the relevant one, $x = r-1$. Thus, the rank $r-1$ under sampling follows a binomial distribution: $r-1 \sim Binomial(n-1, \theta)$, the conditional rank distribution $P(r|R)$ is

$$P(r|R) = Binomial(r-1; n-1, \theta) = \binom{n-1}{r-1}\theta^{r-1}(1-\theta)^{n-r}. \tag{29}$$

Given this, an interesting observation is that the sampling ranked data $\{r_u\}_{r=1}^{M}$ can be directly modeled as a mixture of binomial distributions. Let $\boldsymbol{\Theta} = (\theta_1 \ldots, \theta_R, \ldots, \theta_N)^T$, where

$$\theta_R := \frac{R-1}{N-1}, \quad R = 1, \ldots, N. \tag{30}$$

Let the empirical rank distribution $\tilde{\boldsymbol{P}} = \{\tilde{P}(R)\}_{R=1}^{N}$, then the sampling rank follows the distribution $P(r|\tilde{\boldsymbol{P}}) =$

$$\sum_{R=1}^{N} P(r|R) \cdot P(R) = \sum_{R=1}^{N} Bin(r-1; n-1, \theta_R) \cdot P(R) = \sum_{R=1}^{N} P(R)\binom{n-1}{r-1}\left(\frac{R-1}{N-1}\right)^{r-1}\left(1-\frac{R-1}{N-1}\right)^{n-r}. \tag{31}$$

Thus, $P(R)$ can be considered as the parameters for the mixture of binomial distributions.

### 4.3 Maximum Likelihood Estimation

The basic approach to learning the parameters of the mixture of binomial distributions given $\{r_u\}_{u=1}^{M}$ is based on MLE. Let $\boldsymbol{\Pi} = (\pi_1, \ldots, \pi_R, \ldots, \pi_N)^T$ be the parameters of the mixture of

binomial distributions. Then we have $p(r_u|\mathbf{\Pi}) = \sum_{R=1}^{N} \pi_R \cdot (r_u|\theta_R)$, where $p(r_u|\theta_R) = Binomial(r_u - 1; n-1, \theta_R)$.

Then MLE aims to find the particular $\mathbf{\Pi}$, which maximizes the log-likelihood:

$$\log \mathcal{L} = \sum_{u=1}^{M} \log p(r_u|\mathbf{\Pi}) = \sum_{u=1}^{M} \log \sum_{R=1}^{N} \pi_R p(r_u|\theta_R). \tag{32}$$

By leveraging the EM algorithm (for details, see Appendix D),

$$\pi_R^{new} = \frac{1}{M} \sum_{u=1}^{M} \frac{\pi_R^{old} p(r_u|\theta_R)}{\sum_{j=1}^{N} \pi_j^{old} p(r_u|\theta_j)}. \tag{33}$$

When Equation (33) converges, we obtain $\mathbf{\Pi}^*$ and use it to estimate $\boldsymbol{P}$ (i.e., $\widehat{P}(R) = \pi_R^*$). Then, we can use $\widehat{P}(R)$ in Equation (27) to estimate the desired metric $T$ in Equation (26).

*4.3.1 Speedup and Time Complexity.* To speed the computation, we can further rewrite Equation (33) as follows:

$$\pi_R^{new} = \sum_{r=1}^{n} \tilde{P}(r) \frac{\pi_R^{old} \cdot p(r|\theta_R)}{\sum_{j=1}^{N} \pi_j^{old} \cdot p(r|\theta_j)}, \tag{34}$$

where $\tilde{P}(r) = \frac{1}{M} \sum_{u=1}^{M} \delta(r_u = r)$ is the empirical rank distribution on the sampling data. Thus, the time complexity improves to $O(kNn)$ (from $O(kNM)$ using Equation (33)), where $k$ is the iteration number. This is faster than the least squares solver for the BV estimator (Equation (9)) [24], which is at least $O(n^2N)$. Furthermore, we note $\widehat{P}(R)$ can be used for any metric $T$ for the same algorithm, whereas the BV estimator has to be performed for each metric $T$ separately.

### 4.4 Maximal Entropy with Minimal Distribution Bias (MES)

Another commonly used approach for estimating a (discrete) probability distribution is based on the principle of maximal entropy [3]. Assume a random variable $x$ takes values in $(x_1, x_2, \ldots, x_n)$ with pmf: $p(x_1), p(x_2), \ldots, p(x_n)$. Typically, given a list of (linear) constraints in the form of $\sum_{i=1}^{n} p(x_i) f_k(x_i) \geq F_k$ $(k = 1, \ldots m)$, together with the equality constraint $(\sum_{i=1}^{n} p(x_i) = 1)$, it aims to maximize its entropy $H(p) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$.

In our problem, let the random variable $\mathcal{R}$ take on rank from 1 to $N$. Assume its pmf is $\mathbf{\Pi} = (\pi_1, \ldots, \pi_R, \ldots, \pi_N)$, and the only immediate inequality constraint is $\pi_R \geq 0$ besides $\sum_{R=1}^{N} \pi_R = 1$. Now, to further constrain $\boldsymbol{\pi}$, we need to consider how they reflect and manifest on the observation data $\{r_u\}_{u=1}^{M}$. The natural solution is to simply utilize the (log) likelihood. However, combining them together leads to a rather complex non-convex optimization problem which will complicate the EM solver.

In this article, we introduce a method (to constrain the maximal entropy) that utilizes the squared distance between the learned rank probability (based on $\mathbf{\Pi}$) and the empirical rank probability in the sampling data:

$$\mathcal{E} = \frac{1}{M} \sum_{R=1}^{M} \left( p(r_u|\mathbf{\Pi}) - \tilde{P}(r_u) \right)^2 = \sum_{r=1}^{n} \tilde{P}(r) \left( \sum_{R=1}^{N} P(r|R) \pi_R - \tilde{P}(r) \right)^2. \tag{35}$$

Again, $\tilde{P}(r)$ is the empirical rank distribution in the sampling data. Note that $\mathcal{E}$ can be considered to be derived from the log-likelihood of independent Gaussian distributions if we assume the error term $p(r_u|\mathbf{\Pi}) - \tilde{P}(r_u)$ follows the Gaussian distribution. Given this, we seek to solve the following

optimization problem:

$$\begin{aligned} &\mathbf{\Pi} = \arg\max_{\mathbf{\Pi}} \left( \frac{\eta}{n} \cdot H(\boldsymbol{\pi}) - \mathcal{E} \right) \\ &s.t. \quad \pi_R \geq 0 \ (1 \leq R \leq N), \quad \sum_R \pi_R = 1, \end{aligned} \tag{36}$$

where $\eta$ is the hyperparameter, and $n$ is the sample set size. Note that this objective can also be considered as adding an entropy regularizer for the log-likelihood. The objective function Equation (36) is concave (or its negative is convex). This can be easily observed as both the negative of entropy and the sum of squared errors are convex functions. Given this, we can employ available convex optimization solvers [1] to identify the optimization solution. Thus, we have the estimator $\widehat{P}(R) = \pi^*_R$, where $\Pi^*$ is the optimal solution for Equation (36).

## 5 TOP-*K* METRIC ESTIMATION VIA AN OPTIMIZED BV ESTIMATOR

In this section, we introduce a new estimator that aims to directly minimize the expected errors between the item-sampling-based top-$K$ metrics and the global top-$K$ metrics. Here, we consider a strategy similar to that of Krichene and Rendle [24], although our objective function is different and aims to explicitly minimize the expected error. We aim to search for a *sampled metric* $\widehat{\mathcal{F}}(r)$ to approach $\widehat{T} \approx T$:

$$\widehat{T} = \sum_{r=1}^{n} \tilde{P}(r) \cdot \widehat{\mathcal{F}}(r) = \frac{1}{M} \sum_{u=1}^{M} \widehat{\mathcal{F}}(r_u) \approx \frac{1}{M} \sum_{u=1}^{M} \mathcal{F}(R_u) = \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R) = T,$$

where $\tilde{P}(r) = \frac{1}{M} \sum_{r=1}^{M} \delta(r_u = r)$ is the empirical sampled rank distribution and $\widehat{\mathcal{F}}(r)$ is the adjusted discrete metric function. An immediate observation is this:

$$\mathbb{E}[\widehat{T}] = \sum_{r=1}^{n} \mathbb{E}[\tilde{P}(r)] \cdot \widehat{\mathcal{F}}(r) = \sum_{r=1}^{n} P(r) \cdot \widehat{\mathcal{F}}(r). \tag{37}$$

Following the classical statistical inference [2], the optimality is measured by **Mean Squared Error (MSE)**:

$$\begin{aligned} &\mathbb{E}\left[\widehat{T} - \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R)\right]^2 = \mathbb{E}[\mathbb{E}[T] - \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R) + T - \mathbb{E}[T]]^2 \\ &= \left(\mathbb{E}[T] - \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R)\right)^2 + \mathbb{E}[T - \mathbb{E}[T]]^2 \\ &= \left(\sum_{r=1}^{n} P(r) \cdot \widehat{\mathcal{F}}(r) - \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R)\right)^2 + \mathbb{E}\left[\sum_{r=1}^{n} \tilde{P}(r) \cdot \widehat{\mathcal{F}}(r) - \sum_{r=1}^{n} P(r) \cdot \widehat{\mathcal{F}}(r)\right]^2 \\ &= \left(\sum_{r=1}^{n} \sum_{R=1}^{N} P(r|R) \cdot P(R) \widehat{\mathcal{F}}(r) - \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R)\right)^2 \\ &+ \mathbb{E}\left[\sum_{r=1}^{n} \sum_{R=1}^{N} \tilde{P}(r|R) \cdot P(R) \widehat{\mathcal{F}}(r) - \sum_{r=1}^{n} \sum_{R=1}^{N} P(r|R) \cdot P(R) \widehat{\mathcal{F}}(r)\right]^2. \end{aligned} \tag{38}$$

Remark that $\tilde{P}(r|R)$ is the empirical conditional sampling rank distribution given a global rank $R$. We leverage Jensen's inequality to bound the first term in Equation (38). Specifically, we may treat

$\sum_{r=1}^{n} P(r|R) \cdot \widehat{\mathcal{F}}(r) - \mathcal{F}(R)$ as a random variable and use $(\mathbb{E}X)^2 \leq \mathbb{E}X^2$ to obtain

$$\left(\sum_{r=1}^{n}\sum_{R=1}^{N} P(r|R) \cdot P(R) \cdot \widehat{\mathcal{F}}(r) - \sum_{R=1}^{N} P(R) \cdot \mathcal{F}(R)\right)^2 \leq \sum_{R=1}^{N} P(R)\left(\sum_{r=1}^{n} P(r|R) \cdot \widehat{\mathcal{F}}(r) - \mathcal{F}(R)\right)^2.$$

Therefore, we have

$$\mathbb{E}\left[\widehat{T} - \sum_{R=1}^{N} P(R)\mathcal{F}(R)\right]^2 \leq \underbrace{\sum_{R=1}^{N} P(R)\left\{\left(\sum_{r=1}^{n} P(r|R)\hat{\mathcal{F}}(r) - \mathcal{F}(R)\right)^2\right.}_{\mathcal{L}_1} + \underbrace{\left.\mathbb{E}\left[\sum_{r=1}^{n} \tilde{P}(r|R) \cdot \widehat{\mathcal{F}}(r) - \sum_{r=1}^{n} P(r|R) \cdot \widehat{\mathcal{F}}(r)\right]^2\right\}}_{\mathcal{L}_2}.$$

Let $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$, which gives an upper bound on the expected MSE. Therefore, our goal is to find $\widehat{\mathcal{F}}(r)$ to minimize $\mathcal{L}$. We remark that a seemingly innocent application of Jensen's inequality results in an optimization objective that possesses a range of the following interesting properties.
***(1) Statistical Structure.*** The objective has a variance-bias tradeoff interpretation—that is,

$$\mathcal{L}_1 = \sum_{R=1}^{N} P(R)\left(\mathbb{E}(\widehat{\mathcal{F}}(r)|R) - \mathcal{F}(R)\right)^2, \quad \mathcal{L}_2 = \sum_{R=1}^{N} \frac{1}{M} Var[\widehat{\mathcal{F}}(r)|R], \tag{39}$$

where $\mathcal{L}_1$ can be interpreted as a bias term and $\mathcal{L}_2$ can be interpreted as a variance term. Note that while Krichene and Rendle [24] also introduce a variance-bias tradeoff objective, their objective is constructed from heuristics and contains a hyperparameter (that determines the relative weight between bias and variance) that needs to be tuned in an ad-hoc manner. Here, because our objective is constructed from direct optimization of the MSE, it is more principled and also removes dependencies on hyperparameters. See Section 5.2 for more comparison against estimators proposed in the work of Krichene and Rendle [24].
***(2) Algorithmic Structure.*** Although the objective is not convex, we show that the objective can be expressed in a compact manner using matrices and we can find the optimal solution in a fairly straightforward manner. In other words, Jensen's inequality substantially simplifies the computation at the cost of having a looser upper bound. See Section 5.2.
***(3) Practical Performance.*** Our experiments also confirm that the new estimator is effective (Section 8), which suggests that Jensen's inequality makes only inconsequential and moderate performance impact on the estimator's quality.

### 5.1 Analysis of $\mathcal{L}_2$

To analyze $\mathcal{L}_2$, let us take a close look of $\tilde{P}(r|R)$. Formally, let $X_r$ be the random variable representing the number of items at rank $r$ in the item-sampling data whose original rank in the entire item set is $R$. Then, we rewrite $\tilde{P}(r|R) = \frac{X_r}{M \cdot P(R)}$. Furthermore, it is easy to observe that $(X_1, \ldots X_n)$ follows the multinomial distribution $Multi(P(1|R), \ldots, P(n|R))$.

$$\mathbb{E}[X_r] = M \cdot P(R) \cdot P(r|R), \quad Var[X_r] = M \cdot P(R) \cdot P(r|R)(1 - P(r|R)) \tag{40}$$

Next, let us define a new random variable $\mathcal{B} \triangleq \sum_r^n \widehat{\mathcal{F}}(r)X_r$, which is the weighted sum of random variables under a multinomial distribution. According to Appendix E, its variance is give by

$$Var[\mathcal{B}] = \mathbb{E}\left[\sum_{r=1}^{n} X_r\widehat{\mathcal{F}}(r) - \sum_{r=1}^{n} \mathbb{E}X_r]\widehat{\mathcal{F}}(r)\right]^2 = M \cdot P(R)\Big(\sum_{r}^{n} \widehat{\mathcal{F}}^2(r)P(r|R) - \Big(\sum_{r}^{n} \widehat{\mathcal{F}}(r)P(r|R)\Big)^2\Big).$$

$\mathcal{L}_2$ can be rewritten (see Appendix F) as $\mathcal{L}_2 = \sum_{R=1}^{N} \frac{1}{M} Var[\widehat{\mathcal{F}}(r)|R]$.

### 5.2 Closed-Form Solution and Its Relationship to the BV Estimator

We can rewrite $\mathcal{L}$ as a matrix format and correspond it to a constraint least square optimization (see Appendix G),

$$\mathcal{L} = ||\sqrt{D}A\mathbf{x} - \sqrt{D}\mathbf{b}||_F^2 + \frac{1}{M}||\sqrt{\Lambda_1}\mathbf{x}||_F^2 - \frac{1}{M}||A\mathbf{x}||_F^2, \tag{41}$$

and its solution,

$$\mathbf{x} = \left(A^T DA - \frac{1}{M}A^T A + \frac{1}{M}\Lambda_1\right)^{-1} A^T D\mathbf{b}, \tag{42}$$

where $M$ is the number of users and $diagM(\cdot)$ is a diagonal matrix:

$$\mathbf{x} = \begin{bmatrix} \widehat{\mathcal{F}}(r=1) \\ \vdots \\ \widehat{\mathcal{F}}(r=n) \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathcal{F}(R=1) \\ \vdots \\ \mathcal{F}(R=N) \end{bmatrix} \in \mathbb{R}^N$$

$$A_{R,r} = P(r|R) \in \mathbb{R}^{N\times n}, \quad D = diagM\Big(P(R)\Big) \in \mathbb{R}^{N\times N}, \quad \Lambda_1 = diagM\left(\sum_{R=1}^{N} P(r|R)\right) \in \mathbb{R}^{n\times n}.$$

***Relationship to the BV Estimator.*** The BV estimator is given by Krichene and Rendle [24]:

$$\mathcal{L}_{BV} = \underbrace{\sum_{R=1}^{N} P(R)(\mathbb{E}[\widehat{\mathcal{F}}(r)|R] - \mathcal{F}(R))^2}_{\mathcal{L}_1} + \underbrace{\sum_{R=1}^{N} \gamma \cdot P(R) \cdot Var[\widehat{\mathcal{F}}(r)|R]}_{\mathcal{L}_2}.$$

We observe that the main difference between the BV and our new estimator is on the $\mathcal{L}_2$ components (variance components): for our estimator, each $Var[\widehat{\mathcal{F}}(r)|R]$ is regularized by $1/M$ ($M$ is the number of users), where in BV, this term is regularized by $\gamma \cdot P(R)$ (or $\frac{\gamma}{N}$ if we take uniform distribution $P(R) = \frac{1}{N}$). Our estimator reveals that as the number of users increases, the variance in the $\mathcal{L}_2$ components will continue to decrease, whereas the BV estimator does not consider this factor. Thus, as the user size increases, the BV estimator still needs to deal with $\mathcal{L}_2$ or has to manually adjust $\gamma$.

Finally, both BV and the new estimator rely on prior distribution $P(R)$, which is unknown. In the work of Krichene and Rendle [24], the uniform distribution is used for the estimation purpose. In this article, we propose to leverage the latest approaches in the work of Jin et al. [20], which provide a more reasonable estimation of $P(R)$ for this purpose.

## 6 BOOSTING GLOBAL TOP-*K* METRIC ESTIMATION ACCURACY VIA ADAPTIVE ITEM SAMPLING

### 6.1 Blind Spot Issue and Adaptive Sampling

In recommendation, top-ranked items are vital, and thus it is more crucial to obtain an accurate estimation for these top items. However, current sampling approaches treat all items equally and particularly have difficulty in recovering the global top-$K$ metrics when $K$ is small. In Figure 7, we plot the distribution of target items' rank in the sample set and observe that most target items rank top 1 (highlighted). This could lead to the "blind spot" problem—when $K$ gets smaller, the estimation of basic estimators is more inaccurate (Figure 8). Intuitively, when $r_u = 1$, it does not mean its global rank $R_u$ is 1; instead, its expected global rank may be around 100 (assuming $N = 10K$ and sample set size $n = 100$) according to the analysis in Section 3. And the estimation

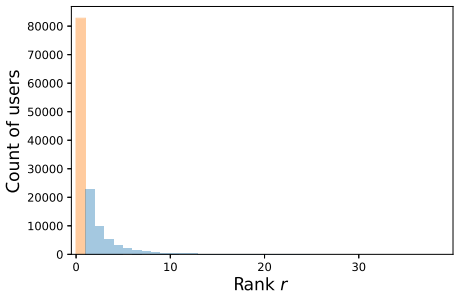


Fig. 7. Distribution of $r_u$ with sample set size $n = 100$. Rank $r = 1$ is highlighted.

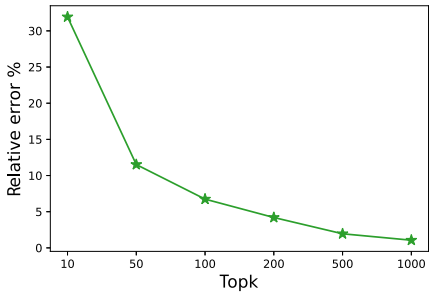


Fig. 8. The relative error of the MLE estimator for different top-$K$. The result is obtained by the EASE model [33] over the *ml-20m* dataset.

**ALGORITHM 1:** Adaptive Sampling Process

**INPUT:** Recommender Model $RS$, test user set $\mathcal{U}$, initial size $n_0$, terminal size $n_{max}$
**OUTPUT:** $\{(u, r_u, n_u)\}$

**for all** $u \in \mathcal{U}$ **do**
  sampling $n_0 - 1$ items, form the sample set $I_u^s$
  $n_u = n_0$, $r_u = RS(i_u, I_u^s)$
  **while** $r_u = 1$ and $n_u \neq n_{max}$ **do**
    sampling extra $n_u$ items, form the new set $I_u^s$
    $n_u = 2n_u$, $r_u = RS(i_u, I_u^s)$
  **end while**
  record $n_u, r_u$ for user $u$
**end for**

granularity is only at around the 1% ($1/n$) level. This blind spot effect brings a big drawback for current estimators.

Based on the preceding discussion, we propose an adaptive sampling strategy, which increases the acceptable test sample size for users whose target item ranks top (say $r_u = 1$) in the sampled data. When $r_u = 1$, we continue doubling the sample size until $r_u \neq 1$ or until the sample size reaches a pre-determined ceiling. See Algorithm 1. Specifically, we start from an initial sample set size parameter $n_0$. We sample $n_0 - 1$ items and compute the rank $r_u$ for all users. For those users with $r_u > 1$, we take down the sample set size $n_u = n_0$. For those with $r_u = 1$, we double the sample set size $n_1 = 2n_0$; in other words, we sample another set of $n_0$ items (since we already sample $n_0 - 1$). Consequently, we check the rank $r_u$ and repeat the process until $r_u \neq 1$ or the sample set size is $n_{max}$. We will discuss how to determine $n_{max}$ later in Section 6.3.

The benefits of this adaptive strategy are twofold: *high granularity*, where with more items sampled, the counts of $r_u = 1$ shall reduce, which could further improve the estimating accuracy; *efficiency*, where we iteratively sample more items for users whose $r_u = 1$ and the empirical experiments (see Table 5) confirm that a small average adaptive sample size (compared to uniform sample size) is able to achieve better performance.

### 6.2 MLE by EM

To utilize the adaptive item sampling for estimating the global top-$K$ metrics, we review two routes: (1) approaches from Krichene and Rendle [24] and our aforementioned estimators in this article;

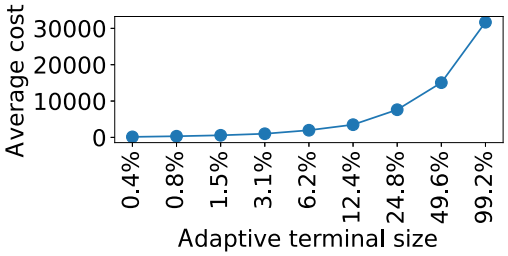


Fig. 9. Sample efficiency w.r.t. terminal size. The illustration result is obtained by the EASE model [33] over the *yelp* dataset while it consistently is observed in other datasets.

(2) methods based on MLE and EM in Section 4. Since every user has different number of item samples, we found that the first route is hard to extend (which requires equal sample size), but luckily the second route is much more flexible and can be easily generalized to this situation.

To begin with, we note that for any user $u$ (his/her test item ranks $r_u$ in the sample set (with size $n_u$) and ranks $R_u$ (unknown)), its rank $r_u$ follows a binomial distribution:

$$P(r = r_u | R = R_u; n_u) = Binomial(r_u - 1; n_u - 1, \theta_u). \tag{43}$$

Given this, let $\mathbf{\Pi} = (\pi_1, \ldots, \pi_R, \ldots, \pi_N)^T$ be the parameters of the mixture of binomial distributions, and $\pi_R$ is the probability for user population ranks at position $R$ globally. And then we have $p(r_u|\mathbf{\Pi}) = \sum_{R=1}^{N} \pi_R \cdot p(r_u|\theta_R; n_u)$, where $p(r_u|\theta_R; n_u) = Bin(r_u - 1; n_u - 1, \theta_R)$. We can apply the MLE to learn the parameters of the mixture of binomial distributions (*MB*), which naturally generalizes the EM procedure (for details, see Appendix H) used in the work of Jin et al. [20], where each user has the same $n$ samples:

$$\phi(R_{uk}) = P(R_u = k | r_u; \boldsymbol{\pi}^{old}), \quad \pi_k^{new} = \frac{1}{M} \sum_{u=1}^{M} \phi(R_{uk}).$$

When the process converges, we obtain $\mathbf{\Pi}^*$ and use it to estimate $\boldsymbol{P}$ (i.e., $\widehat{P}(R) = \pi_R^*$). Then, we can use $\widehat{P}(R)$ in Equation (26) to estimate the desired metric $T$. The overall time complexity is linear with respect to the sample size $O(t \sum n_u)$, where $t$ is the iteration number.

### 6.3 Sampling Size Upper Bound

Now, we consider how to determine the terminal size $n_{max}$. We take the post-analysis over the different terminal sizes and investigate the average sampling cost, which introduces the concept *sampling efficiency* (Figure 9). Formally, we first select a large number $n_{max} \approx N$ and repeat the aforementioned adaptive sampling process. For each user, his/her sampling set size could be one of $\{n_0, n_1 = 2n_0, n_2 = 4n_0, \ldots, n_t = n_{max}\}$. And there are $m_j$ users whose sample set size is $n_j$ $(j = 0, 1, \ldots, t)$. The average sampling cost for each size $n_j$ can be defined heuristically:

$$C_j = \frac{(M - \sum_{p=0}^{j-1} m_p) \times (n_j - n_{j-1})}{m_j} \quad j \neq 0, t \quad C_0 = \frac{M \times n_0}{m_0}. \tag{44}$$

The intuition behind Equation (44) is this: at $j$-th iteration, we independently sample $n_j - n_{j-1}$ items for total $M - \sum_{p=0}^{j-1} m_p$ users, and there are $m_j$ users whose rank $r_u > 1$. $C_j$ is the average items to be sampled to get a user whose $r_u > 1$, which reflects sampling efficiency. In Figure 9, we can see that when the sample reaches 12.4% (of total items, around 3,200 for the *yelp* dataset), the sampling efficiency will reduce quickly (the average cost $C_j$ increases fast). Such post-analysis provides insights on how to balance the sample size and sampling efficiency. In this case, we observe that 12.4% can be a reasonable choice. Even though different datasets can pick up different thresholds,

we found in practice that 10% ∼ 15% can serve as a default choice to start and achieve pretty good performance for the estimation accuracy.

## 7 USER SAMPLING

In real scenarios, the number of users is usually much larger than that of items. For example, there are millions of items in an online shopping portal while the user can be as many as billions. This section examines the sampling effect for the user side. Although sampling users appears to be a natural strategy to speed up evaluation, there has been a lack of study from statistical analysis. Gunawardana and Shani [14] briefly reviewed the approach to compare two models $A$ and $B$ using the sign test [8] and the potentially more sophisticated Wilcoxon signed rank test. However, they do not discuss the statistical nature of the commonly used top-$K$ evaluation metrics based on user sampling and how to use these user-sampling-based metrics to draw a right (statistically rigorous) decision. This work assumes the test dataset can be used for evaluating the performance of recommendation models (where problems like data leakage have been solved [35]). Finally, we underline that this section does not introduce new techniques. Instead, we focus on applying available statistical tools to help quantify user-sampling-based evaluation metrics. Our analysis aims to offer principled guidelines for practitioners in adopting sampling-based approaches to speed up offline evaluation.

### 7.1 Statistical Analysis for One Model

First, we would like to point out that the top-$K$ evaluation metrics on testing data itself are often considered as a special case of user sampling (e.g., the common practice will split the data into 80%-20%). Thus, we hope to use testing user sampling to approximate the overall population:

$$T_{\mathcal{F}@K} = \sum_{R=1}^{K} \tilde{P}(R) \cdot \mathcal{F}(R) \approx \sum_{R=1}^{K} P(R) \cdot \mathcal{F}(R) = \mathbb{E}_R[\mathcal{F}(R)],$$

where $\tilde{P}(R)$ and $P(R)$ are the empirical rank distributions on the testing user population and entire user population, respectively. $\mathcal{F}$ is any metric function (like Recall), and $T_{\mathcal{F}@K}$ is the corresponding metric result.

Let us consider the top-$K$ Recall metric from Equation (2), and it can be written as follows:

$$\frac{1}{M} \sum_{u=1}^{M} \delta(R_u \le K) \triangleq \frac{1}{M} \cdot Q, \tag{45}$$

where the summation is denoted as a symbol $Q$. We assume the $R_u$ for any user $u$ follows the i.i.d. distribution, and thus $\delta(R_u \le K)$ can be treat as a random variable of the Bernoulli distribution with some specific probability $p_K$ such that $\delta(R_u \le K) = 1$. Consequently $Q \sim Binomial(M, p_K)$. This is the widely known point estimation for binomial distributions [2]. When $M$ (number of user) is sufficiently large enough, we assume that the top-$K$ recall metric (a.k.a. $\frac{Q}{M}$) is a good estimator of the underlying probability $p_K$. Clearly, we could estimate the underlying probability with much smaller samples ($m \ll M$), and we can also infer the sampling $m$ with respect to the margin of error $e$:

$$m = p_K(1 - p_K) \cdot \left( \frac{z_{\frac{\alpha}{2}}}{e} \right)^2, \tag{46}$$

where $z_{\frac{\alpha}{2}}$ is the critical value for the corresponding confidence level. One may wonder how to determine the sample size in practice. $Recall@K$ (a.k.a. $p_K$) is between 0 and 1, saying $Recall@30 = 0.3$ for instance. We could also take $p_K = 0.5$ to get the largest sample size. If we set the margin of

error to be 3% and 1% in the 95% confidence level,

$$m = 1.96^2 \cdot \frac{0.5^2}{0.03^2} \approx 1067 \quad m = 1.96^2 \cdot \frac{0.5^2}{0.01^2} \approx 9604. \tag{47}$$

Note that since most of the recommendation models with *Recall@K* (say $K = 50$) is often higher than say 0.4, this can also effectively give us an estimation on the relative error estimation. In fact, this suggests that $10K$ users can be a good rule-of-thumb for user sampling for Recall metrics. In experiment Section 8, we empirically investigate the effect of different user sample sizes.

For the top-$K$ metrics, like AP and NDCG, the individual user metric is always bounded (actually between 0 and 1), then we may adopt Hoeffding's inequality (we can also alternatively use a central limit theorem):

$$Pr\left(\left|\frac{1}{m}\sum_{u}^{m}\left(\delta(R_u \le K)\cdot\mathcal{F}(R_u) - \mathbb{E}[\mathcal{F}(R)]\right)\right| \ge t\right) \le 2\exp(-2mt^2). \tag{48}$$

Given this, we can also infer the sample with a targeted bound of error ($t$). For instance, when $t = 0.02$ (absolute error), and sample size $10K$, the confidence bound is higher than 99.9%.

### 7.2 Statistical Analysis of Multiple Models

First, since there are typically multiple models, the preceding analysis on the sample size and confidence interval analysis should be revised to support that the statistical results for all models hold true. In this case, the Bonferroni correction (or Bonferroni inequality) can be leveraged to remedy this situation. This will lead the sample size to be multiplied. Second, as we need to compare any two models or pick the winners from a list of models, the statistical toolbox would require us to reply on hypothesis testing. For instance, a two-sample $z$-test is used to test the difference between the Recall metrics between two models, which are population proportions $p_1$ and $p_2$,

$$z = \frac{(\hat{p}_1 - \hat{p}_2)}{\sqrt{\frac{2\bar{p}\bar{q}}{m}}}, \tag{49}$$

for the two hypotheses,

$$\begin{aligned} H_0 &: p_1 - p_2 = 0 \\ H_a &: p_1 - p_2 < 0. \end{aligned} \tag{50}$$

Alternatively, we can even derive the sample size based on the confidence interval for $\hat{p}_1 - \hat{p}_2$:

$$(\hat{p}_1 - \hat{p}_2) \pm z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{m_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{m_2}}, \tag{51}$$

where $z_{\frac{\alpha}{2}}$ is the critical value for the standard normal curve with area $C$ between $-z_{\frac{\alpha}{2}}$ and $z_{\frac{\alpha}{2}}$. Setting the sample size $m_1 = m_2 = m$ and the upper bound proportions $\hat{p}_1 = \hat{p}_2 \triangleq p_m = 0.5$, we are able to derive the sample size for a given error range $e$ at specific $C$ confidence:

$$e = z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{2p_m(1-p_m)}{m}}, \quad m = 2p_m(1-p_m)\cdot\left(\frac{z_{\frac{\alpha}{2}}}{e}\right)^2. \tag{52}$$

Compared to the single model described in Section 7.1, the value in Equation (52) is double that of Equation (46).

For the general metrics, like AP and NDCG, which cannot be represented as population proportion, we can resort to the two-sample $t$-test to decide if one model is better than the other. Furthermore, if we consider multiple comparisons at the same time, Bonferroni inequality again has to be used.

Table 2. Dataset Statistics

| Dataset | Interactions | Users | Items | Sparsity |
|---|---|---|---|---|
| *pinterest-20* | 1,463,581 | 55,187 | 9,916 | 99.73% |
| *yelp* | 696,865 | 25,677 | 25,815 | 99.89% |
| *ml-20m* | 9,990,682 | 136,677 | 20,720 | 99.65% |

## 8 EXPERIMENTS

In this section, we investigate the experimental results of mapping function proposed in Section 3, top-$K$ metric estimators proposed in Sections 4 and 5, and also the adaptive estimator in Section 6 and user-based sampling in Section 7. Specifically, we aim to answer the following questions:

- *Q1*: How do various mapping functions $f$ (Section 3.4) help align $T^S_{Recall@K}$ with respect to $T_{Recall@f(K)}$?
- *Q2*: How do these estimators (Sections 4 and 5) perform compared to baseline BV [24] on estimating the top-$K$ metrics based on sampling?
- *Q3*: How effective and efficient is the adaptive item-sampling evaluation method (adaptive MLE (Section 6)) compared with the basic (non-adaptive) item-sampling methods?
- *Q4*: How accurately can these estimators (Sections 4 through 6) find the best model (in terms of the global top-$K$ metric) among a list of recommendation models?
- *Q5*: How effective is the user-sampling-based evaluation method (Section 7)?

### 8.1 Experimental Setting

*8.1.1 Datasets.* We conduct experiments on three widely used relatively large datasets (*pinterest-20, yelp, ml-20m*) for recommendation system research. Table 2 shows the information of these three datasets.

*8.1.2 Recommendation Models.* We use five widely used recommendation algorithms, including three non-deep learning methods (itemKNN [9], ALS [18], and EASE [33]) and two deep learning ones (NeuMF [16] and MultiVAE [29]). The selection of models tries to enable varied performance and advantage in different datasets [24].

*8.1.3 Evaluation Metrics.* The three most popular top-$K$ metrics (Equation (2)): *Recall*, *NDCG*, and *AP* are utilized for evaluating the recommendation models.

*8.1.4 Evaluating and Estimating Procedure.* There are $M$ users and $N$ items. Each user $u$ is associated with a target item $i_u$ (*leave-one-out*). The learned recommendation algorithm/model $A$ would compute the ranks $\{R_u\}_{u=1}^M$ among all items called *global ranks* and the ranks $\{r_u\}_{u=1}^M$ among the sampled item set called *sampled ranks*. Without the knowledge of $\{R_u\}_{u=1}^M$, the estimator tries to estimate the global metric $T$ defined in Equations (1) through (3) based on sampled set test results $\{r_u\}_{u=1}^M$. We repeat experiments 100 times, deriving 100 distinct $\{r_u\}_{u=1}^M$ results. The following reported experimental results are displayed with mean and standard deviation over these 100 repeats.

*8.1.5 Item-Sampling-Based Estimators.* BV (Bias-Variance Estimator) [24]; MLE (Maximal Likelihood Estimation) from Section 4.1 [20]; MES (Maximal Entropy with Squared distribution distance) from Section 4.1 [20]; BV_MLE, BV_MES (Equation (9) with $P(R)$ obtained from MLE and MES, basically, we consider combining the two approaches from BV [24] and MLE/MES [20]); the new multinomial distribution based estimator with different prior, short as MN_MLE, MN_MES, Equation (42) with prior $P(R)$ obtained from MLE and MES estimators.

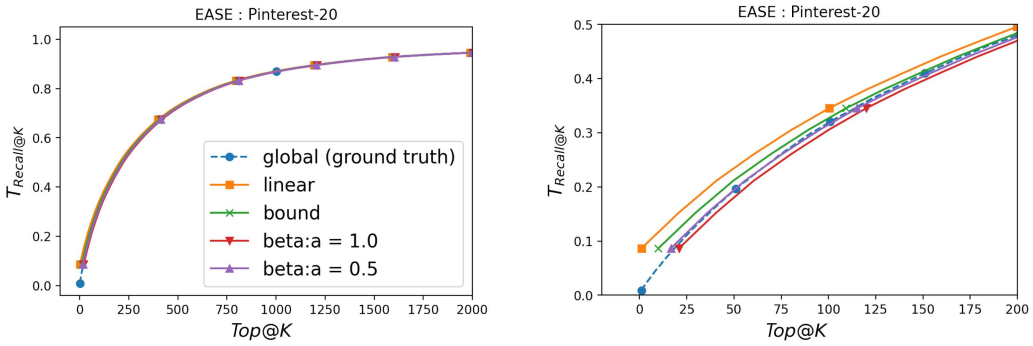


Fig. 10. $T^S_{Recall@k}$ curves alignment with $T_{Recall@K}$ by different mapping function. Left: Results $K$ from 1 to 2,000. Right: Zoom-out showing the details from 1 to 200. All mapping functions exhibit promising approximations, especially bound and $beta@0.5$. An example of the model EASE conducted on the *pinterest-20* dataset, with sample set size $n = 500$.

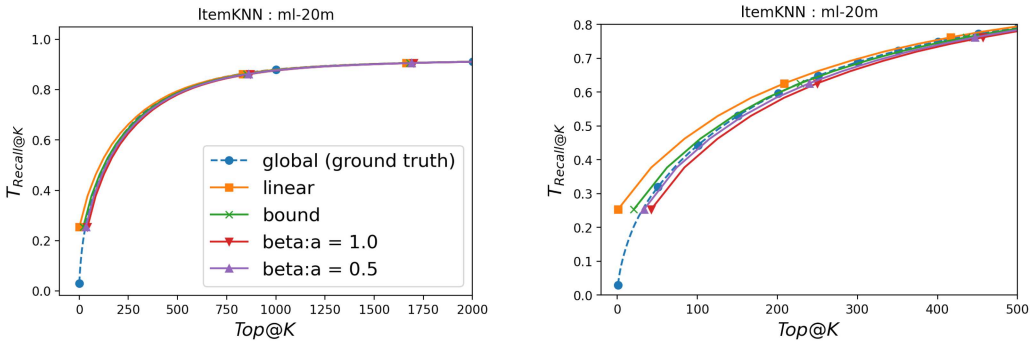


Fig. 11. $T^S_{Recall@k}$ curves alignment with $T_{Recall@K}$ by different mapping function. Left: Results $K$ from 1 to 2,000. Right: Zoom-out showing the details from 1 to 500. All mapping functions exhibit promising approximations, especially bound and $beta@0.5$. An example of model $itemKNN$ conducted on the *ml-20m* dataset, wi8th sample set size $n = 500$.

*8.1.6 Reproducibility.* The source code is available at https://github.com/dli12/Item-Sampling-Recommendation-Evaluation/.

## 8.2 Aligning Sampling and the Global Top-*K* Recall Metric (Q1)

This section provides a holistic view of the alignment between sampling and global top-$K$ Recall (curves). As we discussed in Section 3.4, there exists a mapping function $f$ such that $T^S_{Recall@K} \approx T_{Recall@f(K)}$. Thus, for each point $(K, T^S_{Recall@K})$ in the sampled Recall curve, it can be treated as a point $(f(K), T^S_{Recall@K})$ in a global scale. We then plot these global scale curves (generated by different mapping functions) together with the original global Recall curve in Figure 10 and Figure 11. Here, we report two examples with four different approximating mapping functions from Section 3.4, the linear, bound, $beta@1$, and $beta@0.5$, for the curve alignment. We observe from the figures that both bound and $beta@0.5$ achieve superior results, where they are closest to the ground truth curve, which further validates our claim of mapping functions.

We can evaluate the effectiveness of function mapping by measuring winner prediction. For instance, suppose we consider the first row of Table 3 and use the *Beta (a = 0.5)* mapping function to map $K = 1$ to $f(K) = 9$. This implies that the value of the actual global Recall metric at top-9

Table 3. Winner Prediction Ability of Different Mapping Functions

| Sampled Top-$K$ | Dataset | Mapping Function | | | |
|---|---|---|---|---|---|
| | | Linear | Bound | Beta ($a$ = 0.5) | Beta ($a$ = 1) |
| $K$ = 1 | *pinterest-20* | $f(K) = 1$ | $f(K) = 5$ | $f(K) = 9$ | $f(K) = 11$ |
| | | 11 | 89 | 89 | 89 |
| | *yelp* | $f(K) = 1$ | $f(K) = 13$ | $f(K) = 21$ | $f(K) = 27$ |
| | | 0 | 100 | 100 | 100 |
| | *ml-20m* | $f(K) = 1$ | $f(K) = 10$ | $f(K) = 17$ | $f(K) = 22$ |
| | | 100 | 100 | 100 | 100 |
| $K$ = 2 | *pinterest-20* | $f(K) = 11$ | $f(K) = 15$ | $f(K) = 19$ | $f(K) = 21$ |
| | | 90 | 90 | 90 | 90 |
| | *yelp* | $f(K) = 27$ | $f(K) = 39$ | $f(K) = 47$ | $f(K) = 53$ |
| | | 100 | 100 | 100 | 100 |
| | *ml-20m* | $f(K) = 22$ | $f(K) = 31$ | $f(K) = 38$ | $f(K) = 42$ |
| | | 100 | 100 | 100 | 100 |

Sample set size $n = 1{,}000$.

($T_{Recall@9}$) is similar to the value of the sampled metric at top-1 ($T^S_{Recall@1}$). By comparing the recommendation models (e.g., itemKNN, ALS, NeuMF, MultiVAE, EASE), we determine the best model in terms of both $T^S_{Recall@1}$ and $T_{Recall@9}$. We perform 100 repeated experiments on different sampled sets and record the number of times the sampled Recall metric is consistent with the global Recall metric, with 89 successful trials observed in this example. As shown in Table 3 and Figures 10 and 11, the mapping functions typically assign the sampled Recall metric to moderate positions that accurately reflect the relative order of the models. It is worth noting that given our primary interest in the top positions (approximately $K\frac{N}{n}$ after mapping) of a metric, we set the sample size $n$ to a moderately large value of 1,000 and keep $K$ relatively small ($K = 1, 2$).

### 8.3 Estimation Accuracy of Estimators (Q2)

Here, we aim to answer Question 2: how do these estimators proposed in this article perform compared to baseline BV [24] on estimating the top-$K$ metrics based on sampling? Here, we would try to quantify the accuracy of each estimator in terms of relative error, leading to a more rigorous and reliable comparison. Specifically, we compute the true global $T_{metric@K}$ ($K$ from 1 to 50), then we average the absolute relative error between the estimated $\hat{T}_{metric@k}$ from each estimator and the true one.

Similar to Krichene and Rendle [24], for $\gamma$ in the BV estimator, we tune from $\{1, 0.1, 0.01, 0.001\}$ and $\gamma = 0.01$ is presented as the best ones for overall datasets. For $\eta$ in MES, we tune in the same way and the result of $\eta = 0.01$ is presented. Table 4 presents the average relative error of the estimators in terms of $T_{Recall@K}$ ($k$ from 1 to 50). We highlight the most and the second-most accurate estimator. For instance, for model EASE in dataset *pinterest-20* (line 1 of Table 4), the estimator MN_MES is the most accurate one with 5.00% average relative error compared to its global $T_{Recall@K}$ ($K$ from 1 to 50).

Overall, we observe from Table 4 that MN_MES and MN_MLE are among the most or the second-most accurate estimators. And in most cases, they outperform the others significantly. Meanwhile, they have a smaller deviation compared to their prior estimators MES and MLE. In addition, we notice that the estimators with the knowledge of some reasonable prior distribution (BV_MES, MN_MES, BV_MLE, MN_MLE) could achieve more accurate results than the others. This indicates that these estimators could better help the distribution converge.

### 8.4 Efficiency and Effectiveness of the Adaptive Estimator (Q3)

Here, we aim to answer Question 3. Table 5 presents the average relative error of the estimators in terms of $T_{Recall@K}$ ($k$ from 1 to 50). We highlight the most accurate estimator. For the basic item

Table 4. Average Relative Errors between Estimated $\widehat{T}_{Recall@K}$ ($K$ from 1 to 50) and the True Ones $T_{Recall@K}$

| Dataset | Model | Sample Set Size 100 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | MES | MLE | BV | BV_MES | BV_MLE | MN_MES | MN_MLE |
| *pinterest-20* | EASE | 5.86±2.26 | 5.54±1.85 | 8.11±2.00 | **5.05±1.46** | 5.14±1.46 | **5.00±1.39** | 5.10±1.34 |
| | MultiVAE | 4.17±2.91 | 3.34±2.07 | **2.75±1.61** | 2.89±1.74 | **2.88±1.74** | **2.75±1.66** | **2.75±1.68** |
| | NeuMF | 5.17±2.74 | 4.28±1.95 | 4.23±1.79 | 3.83±1.59 | 3.84±1.72 | **3.60±1.50** | **3.76±1.44** |
| | itemKNN | 5.90±2.20 | 5.80±1.60 | 8.93±1.70 | **5.11±1.22** | 5.31±1.25 | **5.09±1.15** | 5.26±1.14 |
| | ALS | 4.19±2.37 | 3.44±1.68 | 3.17±1.34 | 3.05±1.39 | 3.07±1.42 | **2.86±1.27** | **2.90±1.28** |
| *yelp* | EASE | 8.08±4.94 | 7.89±4.70 | 18.60±2.78 | 6.10±3.74 | 6.56±3.90 | **4.84±2.17** | **5.61±2.30** |
| | MultiVAE | 9.33±6.61 | 7.67±4.94 | 9.70±3.22 | 6.84±4.10 | 6.80±4.04 | **4.30±1.27** | **4.35±1.31** |
| | NeuMF | 15.09±6.24 | 15.47±5.55 | 22.40±3.17 | **13.14±4.55** | 13.92±4.70 | **13.46±2.43** | 14.50±2.45 |
| | itemKNN | 9.25±4.87 | 9.62±4.88 | 23.24±2.16 | **7.69±4.09** | 8.15±4.17 | **7.74±2.08** | 8.75±2.08 |
| | ALS | 14.31±3.96 | 13.68±3.51 | 15.14±1.86 | 13.43±3.16 | 13.26±3.08 | **11.68±0.88** | **11.57±0.83** |
| *ml-20m* | EASE | 10.45±1.03 | 11.52±1.03 | 36.59±0.31 | **8.99±0.74** | 9.86±0.77 | **9.07±0.61** | 10.09±0.69 |
| | MultiVAE | 9.93±0.38 | **9.48±0.22** | 22.24±0.37 | 9.85±0.36 | **9.50±0.22** | 9.82±0.28 | 9.53±0.14 |
| | NeuMF | 4.35±1.50 | 6.05±1.35 | 28.27±0.42 | **3.67±1.14** | 4.81±1.14 | **3.64±1.05** | 4.79±1.08 |
| | itemKNN | 15.31±1.18 | 17.19±1.15 | 36.63±0.42 | **14.02±0.75** | 15.24±0.83 | **14.16±0.68** | 15.41±0.77 |
| | ALS | 36.17±0.83 | **35.21±0.64** | 36.39±0.21 | 36.50±0.74 | 35.75±0.62 | 36.32±0.56 | **35.60±0.48** |

Unit is percentage (%). In each row, the smallest two results are highlighted in bold, indicating the most accurate results. Sample set size $n = 100$.

Table 5. Comparison of Adaptive Estimators with Basic Ones in Terms of *Recall*

| Dataset | Models | Fix Sample | | | | Adaptive Sample | |
|---|---|---|---|---|---|---|---|
| | | BV_MES | BV_MLE | MN_MES | MN_MLE | average size | adaptive MLE |
| sample set size $n = 500$ | | | | | | | |
| *pinterest-20* | EASE | 2.54±0.85 | 2.68±0.87 | 2.78±1.05 | 2.83±1.06 | 307.74±1.41 | **1.69±0.60** |
| | MultiVAE | 2.17±1.08 | 2.13±1.09 | 2.60±1.30 | 2.55±1.35 | 286.46±1.48 | **1.95±0.65** |
| | NeuMF | 2.45±1.15 | 2.44±1.15 | 2.76±1.37 | 2.80±1.38 | 259.77±1.28 | **2.00±0.81** |
| | itemKNN | 2.49±0.97 | 2.59±0.94 | 2.79±1.12 | 2.79±1.20 | 309.56±1.31 | **1.63±0.51** |
| | ALS | 2.65±1.04 | 2.63±1.06 | 3.02±1.32 | 2.98±1.33 | 270.75±1.22 | **2.00±0.73** |
| sample set size $n = 500$ | | | | | | | |
| *yelp* | EASE | 4.68±2.43 | 4.56±2.35 | **3.47±1.79** | 3.49±1.78 | 340.79±2.03 | 3.48±1.40 |
| | MultiVAE | 6.14±3.48 | 6.07±3.46 | 4.68±2.27 | **4.67±2.28** | 288.70±2.24 | 5.08±2.14 |
| | NeuMF | 6.59±2.38 | 6.73±2.35 | 5.48±1.43 | 5.68±1.42 | 290.62±2.11 | **4.01±1.51** |
| | itemKNN | 3.94±1.94 | 3.95±1.92 | **2.92±1.60** | 2.96±1.57 | 369.16±2.51 | 3.25±1.59 |
| | ALS | 10.00±3.47 | 10.31±3.65 | 9.29±2.03 | 9.80±2.23 | 297.07±2.29 | **5.25±2.38** |
| sample set size $n = 1000$ | | | | | | | |
| *ml-20m* | EASE | 1.39±0.21 | 1.69±0.28 | 1.81±0.46 | 1.73±0.46 | 899.89±1.90 | **1.07±0.24** |
| | MultiVAE | 2.23±0.58 | 2.91±0.72 | 3.55±1.23 | 2.98±1.50 | 771.26±1.84 | **1.10±0.39** |
| | NeuMF | 0.82±0.30 | 0.85±0.28 | 1.51±0.66 | 1.69±0.70 | 758.45±1.61 | **0.78±0.27** |
| | itemKNN | 1.84±0.24 | 2.13±0.27 | 1.97±0.42 | 2.17±0.49 | 725.72±1.49 | **1.17±0.28** |
| | ALS | 9.41±0.97 | 12.83±1.27 | 10.63±2.53 | 10.57±3.18 | 705.76±1.56 | **4.29±1.05** |

The average relative errors between estimated $\widehat{T}_{Recall@K}$ ($K$ from 1 to 50) and the true ones. Unit is percentage (%). In each row, the smallest relative error is highlighted, indicating the most accurate result.

sampling, we choose 500 sample size for the datasets *pinterest-20* and *yelp*, and 1,000 sample size for the dataset *ml-20m*. The upper bound threshold $n_{max}$ is set at 3,200.

We observe that adaptive sampling uses much less sample size (typically 200 $\sim$ 300 vs 500 on the first two datasets and 700 $\sim$ 800 vs 1,000 on the last dataset). In particular, the relative error of the adaptive sampling is significantly less than that of the basic sampling methods. On the first (*pinterest-20*) and third (*ml-20m*) datasets, the relative errors have reduced to less than 2%. In other words, the adaptive method has been much more effective (in terms of accuracy) and efficient (in terms of sample size). This also confirms the benefits in addressing the "blind spot" issue, which provides higher resolution to recover global $K$ metrics for small $K$ ($K \leq 50$ here).

Table 6. Accuracy of Estimating the Winner (of the Recommendation Models)

| Dataset | Top-$K$ | Metric | Fix Sample | | | | | | | Adaptive Sample |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MES | MLE | BV | BV_MES | BV_MLE | MN_MES | MN_MLE | adaptive MLE |
| | | | sample set size $n$ = 500 | | | | | | | size 260∼310 |
| *pinterest-20* | 5 | RECALL | 53 | 56 | 61 | 54 | 57 | 53 | 53 | **69** |
| | | NDCG | 51 | 54 | 60 | 52 | 54 | 51 | 52 | **71** |
| | | AP | 51 | 53 | 58 | 51 | 53 | 51 | 51 | **60** |
| | 10 | RECALL | 66 | 66 | 69 | 69 | 73 | 67 | 69 | **78** |
| | | NDCG | 55 | 58 | 65 | 58 | 59 | 58 | 60 | **84** |
| | | AP | 53 | 55 | 61 | 54 | 57 | 52 | 52 | **68** |
| | 20 | RECALL | 69 | 69 | 75 | 69 | 73 | 70 | 74 | **81** |
| | | NDCG | 69 | 69 | 78 | 69 | 73 | 68 | 73 | **79** |
| | | AP | 55 | 58 | 62 | 57 | 60 | 54 | 56 | **69** |
| | | | sample set size $n$ = 500 | | | | | | | size 280∼370 |
| *yelp* | 5 | RECALL | 75 | 94 | 97 | 95 | 94 | 100 | 100 | 96 |
| | | NDCG | 73 | 89 | 97 | 95 | 94 | 100 | 100 | 84 |
| | | AP | 71 | 87 | 97 | 94 | 94 | 100 | 100 | 80 |
| | 10 | RECALL | 88 | 95 | 100 | 98 | 97 | 100 | 100 | **100** |
| | | NDCG | 82 | 94 | 98 | 96 | 96 | 100 | 100 | **100** |
| | | AP | 76 | 94 | 97 | 95 | 95 | 100 | 100 | 94 |
| | 20 | RECALL | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | NDCG | 94 | 98 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | AP | 82 | 94 | 99 | 97 | 96 | 100 | 100 | 98 |
| | | | sample set size $n$ = 1,000 | | | | | | | size 700∼900 |
| *ml-20m* | 5 | RECALL | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | NDCG | 96 | 100 | 100 | 100 | 100 | 98 | 98 | **100** |
| | | AP | 91 | 100 | 100 | 100 | 100 | 96 | 96 | **100** |
| | 10 | RECALL | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | NDCG | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | AP | 98 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | 20 | RECALL | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | NDCG | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |
| | | AP | 100 | 100 | 100 | 100 | 100 | 100 | 100 | **100** |

Values in the table are the number of corrects that predict the winner out of 100 repeat tests. We highlight the adaptive estimator if it achieves the best performance w.r.t. each metric (each row).

### 8.5 Estimating Winner of Recommender Models (Q4)

Besides the estimation accuracy, we also care about whether the estimator can correctly find the best recommendation model. Initially, the reason we compare the performance of various recommendation models (among EASE, ALS, itemKNN, NeuMF, MultiVAE in our work) is to try to find the best model(s) to deploy. The best model is determined by the global top-$K$ metric in Equation (3) ($\{R_u\}$). Thus, it is also meaningful to validate whether our estimated metric $\widehat{T}$ could find the correct "winner" (best model) as the original metric $T$.

Table 6 indicates the results of among the 100 repeats, how many times an estimator could find the best recommendation algorithm for a given metric (Recall, NDCG, AP). We have the following observations. First, the adaptive estimator could achieve the best accuracy in most cases while costing less on average, which enhances its validity. Second, we notice all estimators obtain good results in the *ml-20m* dataset due to its large sample size of $n$ = 1,000. Third, as for the *yelp* dataset with the sample size $n$ = 500, we notice that new proposed expected estimators MN_MES and MN_MLE achieve perfect results while the baseline BV estimator can also obtain comparable results. BV_MES, BV_MLE, MN_MES, MN_MLE all have better results than their prior MES, MLE, further indicating that estimators with reasonable prior could better help the distribution (of the prior) to converge. Fourth, the majority of the estimators did not exhibit good performance in the *pinterest-20* dataset. Upon closer examination, as illustrated in Figure 12, we observed that the performances of the top two models, itemKNN and EASE, are remarkably similar, with differences less than 1e-3 for all top-50 metrics. Such a marginal difference is too subtle for estimators

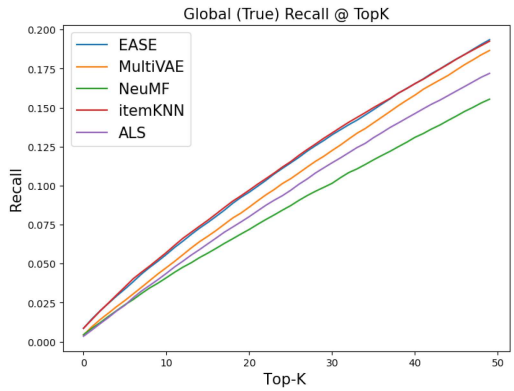


Fig. 12. Global *Recall@K* for all models.

Table 7. Accuracy of Estimating the Top-2 Models in the *pinterest-20* Dataset

| Top-$K$ | Metric | sample size $n$ = 100 | | | | | | | sample size $n$ = 500 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MES | MLE | BV | BV_MES | BV_MLE | MN_MES | MN_MLE | MES | MLE | BV | BV_MES | BV_MLE | MN_MES | MN_MLE |
| 5 | RECALL | 76 | 89 | 93 | 98 | 95 | 99 | 99 | 97 | 100 | 100 | 100 | 100 | 100 | 100 |
| | NDCG | 75 | 89 | 93 | 98 | 95 | 99 | 98 | 90 | 100 | 100 | 100 | 100 | 100 | 99 |
| | AP | 74 | 89 | 93 | 98 | 95 | 99 | 98 | 83 | 100 | 100 | 100 | 100 | 99 | 99 |
| 10 | RECALL | 81 | 89 | 93 | 98 | 98 | 99 | 99 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | NDCG | 79 | 89 | 93 | 98 | 96 | 99 | 99 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | AP | 76 | 89 | 93 | 98 | 95 | 99 | 99 | 93 | 100 | 100 | 100 | 100 | 100 | 100 |
| 20 | RECALL | 89 | 92 | 96 | 99 | 99 | 99 | 99 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |
| | NDCG | 83 | 90 | 94 | 99 | 98 | 99 | 99 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | AP | 79 | 89 | 93 | 98 | 98 | 99 | 99 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |

to effectively discern. In addition, if two models yield almost identical performance, there is little rationale for distinguishing between them (when the model computational cost is not considered). Considering this case, we examine whether estimators can successfully predict the top-2 models (Table 7). In general, as the sample size increased from 100 to 500, the performance of all estimators improved. Notably, with a smaller sample size of $n = 100$, the estimators (BV_MES, BV_MLE, MN_MES, MN_MLE) performed well, with BV also demonstrating comparable results.

### 8.6 Effectiveness of User Sampling (Q5)

In this subsection, we empirically show the results of user sampling. Tables 8 and 9 compare the user sampling method with the estimator MN_MES and the adaptive estimator in terms of $T_{Recall@K}$ and $T_{NDCG@K}$ ($K$ from 1 to 50). In general, we could conclude that even with a small portion of users (e.g.,, 1,000 (0.7%) sampled users compared to its total 137,000 users for the *ml-20m* dataset), the user-sampling-based method could achieve pretty accurate results (4% $\sim$ 8% relative errors for top-$K$ from 1 to 50). In addition, as the size of sampled user increase, it could be significantly close to the true results. For instance, with 10,000 (7%) users sampled for the *ml-20m* dataset, it can achieve as small as 1% relative errors. This consistent empirical results together with the demo example in Equation (47) indicate the effectiveness of user sampling. Noting that, according to Equation (46), the accuracy is not quite related to user size $M$, which suggests that for some very huge dataset (e.g., $M >> 1$ million and $M >> N$, which is quite common in e-commerce), user-sampling based estimation can be more practical and fundamentally efficient than item-sampling-based estimation.

Table 8. Average Relative Errors between Estimated $\hat{T}_{Recall@K}$ ($K$ from 1 to 50) and the Ground Truth

| Dataset | Model | RECALL | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | item 100 | item 500 | adaptive size | adaptive MLE | user 1K | user 5K | user 10K |
| *pinterest-20* | EASE | 5.00±1.39 | 2.78±1.05 | 307.74±1.41 | 1.69±0.60 | 9.04±4.32 | 3.85±1.76 | 2.65±1.33 |
| | MultiVAE | 2.75±1.66 | 2.60±1.30 | 286.46±1.48 | 1.95±0.65 | 9.54±4.41 | 4.34±1.94 | 3.18±1.35 |
| | NeuMF | 3.60±1.50 | 2.76±1.37 | 259.77±1.28 | 2.00±0.81 | 10.43±4.56 | 4.66±2.17 | 3.11±1.39 |
| | itemKNN | 5.09±1.15 | 2.79±1.12 | 309.56±1.31 | 1.63±0.51 | 8.91±4.27 | 3.64±1.52 | 2.65±1.22 |
| | ALS | 2.86±1.27 | 3.02±1.32 | 270.75±1.22 | 2.00±0.73 | 10.24±4.96 | 4.19±2.05 | 3.25±1.39 |
| *yelp* | EASE | 4.84±2.17 | 3.47±1.79 | 340.79±2.03 | 3.48±1.40 | 12.21±5.98 | 4.97±2.16 | 3.62±1.84 |
| | MultiVAE | 4.30±1.27 | 4.68±2.27 | 288.70±2.24 | 5.08±2.14 | 15.70±6.78 | 6.58±2.51 | 4.45±1.83 |
| | NeuMF | 13.46±2.43 | 5.48±1.43 | 290.62±2.11 | 4.01±1.51 | 12.45±6.75 | 5.69±2.96 | 4.23±2.00 |
| | itemKNN | 7.74±2.08 | 2.92±1.60 | 369.16±2.51 | 3.25±1.59 | 11.62±6.01 | 4.59±1.77 | 3.33±1.63 |
| | ALS | 11.68±0.88 | 9.29±2.03 | 297.07±2.29 | 5.25±2.38 | 15.36±6.79 | 6.54±2.18 | 4.46±1.71 |
| *ml-20m* | EASE | 9.07±0.61 | 2.31±0.44 | 899.89±1.90 | 1.07±0.24 | 4.48±2.07 | 1.92±0.91 | 1.33±0.68 |
| | MultiVAE | 9.82±0.28 | 4.60±0.99 | 771.26±1.84 | 1.10±0.39 | 5.97±2.57 | 2.49±1.04 | 1.80±0.73 |
| | NeuMF | 3.64±1.05 | 1.63±0.82 | 758.45±1.61 | 0.78±0.27 | 5.58±2.35 | 2.35±1.10 | 1.67±0.80 |
| | itemKNN | 14.16±0.68 | 3.61±0.61 | 725.72±1.49 | 1.17±0.28 | 5.27±2.73 | 2.21±1.11 | 1.54±0.76 |
| | ALS | 36.32±0.56 | 19.33±1.93 | 705.76±1.56 | 4.29±1.05 | 7.70±2.92 | 3.13±1.13 | 2.22±0.85 |

Unit is percentage (%). $item100$ and $item500$ are the results of item-sampling-based estimator MN_MES with sample set size $n = 100$ and $n = 500$. $user1K$ and so forth are the results of user-sampling-based unbiased average estimation (from Section 7) with $1K$ users sampled.

Table 9. Average Relative Errors between Estimated $\widehat{T}_{NDCG@K}$ ($K$ from 1 to 50) and the Ground Truth

| Dataset | Model | NDCG | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | item 100 | item 500 | adaptive size | adaptive MLE | user 1K | user 5K | user 10K |
| *pinterest-20* | EASE | 9.35±3.09 | 4.17±2.45 | 307.74±1.41 | 1.46±0.63 | 11.02±6.81 | 4.21±2.72 | 3.05±1.99 |
| | MultiVAE | 3.13±2.08 | 3.26±2.14 | 286.46±1.48 | 1.67±0.70 | 10.48±6.52 | 4.88±2.90 | 3.68±2.06 |
| | NeuMF | 4.27±2.44 | 3.24±2.30 | 259.77±1.28 | 1.73±0.83 | 11.99±6.56 | 5.13±3.11 | 3.55±2.04 |
| | itemKNN | 9.69±2.74 | 4.23±2.47 | 309.56±1.31 | 1.42±0.67 | 10.46±6.72 | 3.92±2.42 | 2.96±1.68 |
| | ALS | 3.70±2.00 | 3.90±2.24 | 270.75±1.22 | 1.84±1.07 | 11.29±7.46 | 4.54±3.06 | 3.57±1.98 |
| *yelp* | EASE | 5.36±2.40 | 4.03±2.53 | 340.79±2.03 | 3.55±2.00 | 12.83±8.00 | 5.56±3.32 | 4.02±2.74 |
| | MultiVAE | 4.31±1.90 | 5.77±3.87 | 288.70±2.24 | 5.09±2.60 | 16.69±9.69 | 7.37±4.14 | 4.77±2.62 |
| | NeuMF | 22.50±2.33 | 8.43±4.07 | 290.62±2.11 | 4.43±2.55 | 14.24±9.33 | 6.78±4.86 | 5.08±3.35 |
| | itemKNN | 10.53±2.14 | 3.65±2.28 | 369.16±2.51 | 3.67±2.73 | 12.79±7.81 | 4.83±2.66 | 3.56±2.54 |
| | ALS | 16.91±3.33 | 12.57±5.46 | 297.07±2.29 | 5.48±3.34 | 16.10±9.03 | 6.91±3.06 | 4.60±2.37 |
| *ml-20m* | EASE | 18.98±0.89 | 5.59±1.49 | 899.89±1.90 | 2.01±0.56 | 4.94±3.27 | 2.24±1.40 | 1.57±1.04 |
| | MultiVAE | 16.28±1.56 | 7.01±2.18 | 771.26±1.84 | 1.21±0.60 | 6.31±3.64 | 2.77±1.55 | 1.90±1.09 |
| | NeuMF | 5.67±1.24 | 2.10±1.31 | 758.45±1.61 | 0.92±0.51 | 5.74±3.26 | 2.68±1.80 | 1.91±1.27 |
| | itemKNN | 28.66±1.00 | 7.40±1.60 | 725.72±1.49 | 2.02±0.63 | 6.04±4.00 | 2.43±1.60 | 1.70±1.17 |
| | ALS | 52.19±2.50 | 26.31±3.81 | 705.76±1.56 | 5.27±1.39 | 7.99±3.90 | 3.26±1.59 | 2.33±1.20 |

Unit is percentage (%). $item100$ and $item500$ are the results of item-samplingbased estimator MN_MES with sample set size $n = 100$ and $n = 500$. $user1K$ and so forth are the results of user-sampling-based unbiased average estimation (from Section 7) with $1K$ users sampled.

## 9 CONCLUSION

In this article, we holistically discussed the story of sampling-based top-$K$ recommendation evaluation. Starting from the "inconsistent" phenomenon that was first discovered in the work of Krichene and Rendle [24] and Rendle [32], we [26] observed and proposed the alignment theory in terms of the Recall metric in Section 3. Then we proposed two estimators, MES and MLE, in Section 4, which not only estimate the global user rank distribution $P(R)$ but also help estimate the global true metric [27]. Consequently, we proposed item-sampling estimators in Section 5, which explicitly optimize its MSE with respect to the ground truth. We highlighted the subtle difference between the estimators from Krichene and Rendle [24] and ours, and pointed out the potential issue of the former—failing to link the user size with the variance [27]. Furthermore, we addressed the

limitations of the current item-sampling approaches, which typically do not have sufficient granularity to recover the top-$K$ global metrics when $K$ is small. We then proposed an effective adaptive item-sampling method in Section 6. We also discussed another sampling evaluation strategy from the perspective of user sampling evaluation in Section 7. The experimental results validated the effectiveness of the estimators. Our results provided a solid step toward making both item sampling and user sampling available for recommendation research and practice.

## APPENDICES

## A PROOF OF SAMPLING THEOREM

Proof. Recall Equation (14): $\mathbb{E}[T^S_{Recall@K}] = \sum_{R=1}^{N} \tilde{P}(R) \cdot P(R) = \sum_{u=1}^{M} Pr(r_u \le K)$. Let us assign each user $u$ the weight $Pr(r_u \le K)$ for both curves, $T^{(1)}_{Recall@K}$ and $T^{(2)}_{Recall@K}$. Now, let us build a bipartite graph by connecting any $u$ in the $T^{(1)}_{Recall@K}$ with user $v$ in $T^{(2)}_{Recall@K}$, if $R_u \le R_v$. We can then apply Hall's marriage theorem to claim there is a one-to-one matching between users in $T^{(1)}_{Recall@K}$ to users in $T^{(2)}_{Recall@K}$, such that $R_u \le R_v$, and $Pr(r_u \le K) \ge Pr(r_v \le K)$. (To see that, use the fact that $\sum_{R=1}^{K} \tilde{P}^{(1)}(R) \ge \sum_{R=1}^{K} \tilde{P}^{(2)}(R)$, where $\tilde{P}^{(1)}(R)$ and $\tilde{P}^{(2)}(R)$ are the empirical probability mass distributions of user ranks, or equivalently, $\sum_{R=K}^{N} \tilde{P}^{(1)}(R) \le \sum_{R=K}^{N} \tilde{P}^{(2)}(R)$). Thus, any subset in $T^{(1)}_{Recall@K}$ is always smaller than its neighbor set $N(T^{(1)}_{Recall@K})$ in $T^{(2)}_{Recall@K}$. Given this, we can observe that the theorem holds. □

## B DERIVATION OF BETA DISTRIBUTION APPROXIMATION

The left term of Equation (23) is denoted as $\mathcal{L}_k$:

$$\begin{aligned}
\mathcal{L}_k &= \sum_{R=1}^{N} \tilde{P}(R) \cdot \delta(R \le f(k)) = \sum_{R=1}^{f(k)} \tilde{P}(R) = \frac{1}{\mathcal{B}(a,1)} \sum_{R=1}^{f(k)} \left(\frac{R-1}{N-1}\right)^{a-1} \cdot \frac{1}{N-1} \\
&= \frac{1}{\mathcal{B}(a,1)} \sum_{x=0}^{\frac{f(k)-1}{N-1}} x^{a-1} \cdot \Delta x \quad \text{where}, x = \frac{R-1}{N-1}, \Delta x = \frac{1}{N-1}, \\
&\approx \frac{1}{\mathcal{B}(a,1)} \int_0^{\frac{f(k)-1}{N-1}} x^{a-1} dx = \frac{1}{a\mathcal{B}(a,1)} \left[\frac{f(k)-1}{N-1}\right]^a.
\end{aligned}$$

Considering sampling with replacement, then the right term of Equation (23) is denoted as follows:

$$\mathcal{R}_k = \sum_{i=0}^{k-1} \binom{n-1}{i} \sum_{R=1}^{N} \tilde{P}(R) \left(\frac{R-1}{N-1}\right)^i \left(1 - \frac{R-1}{N-1}\right)^{n-i-1}.$$

Calculate the difference:

$$\begin{aligned}
\mathcal{R}_{k+1} - \mathcal{R}_k &= \binom{n-1}{k} \sum_{R=1}^{N} \tilde{P}(R) \left(\frac{R-1}{N-1}\right)^k \left(1 - \frac{R-1}{N-1}\right)^{n-1-k} \approx \binom{n-1}{k} \frac{1}{\mathcal{B}(a,1)} \int_{x=0}^{1} x^{a+k-1} (1-x)^{n-1-k} dx \\
&= \binom{n-1}{k} \frac{1}{\mathcal{B}(a,1)} \mathcal{B}(a+k, n-k) = \frac{1}{\mathcal{B}(a,1)} \frac{\Gamma(n)}{\Gamma(n+a)} \frac{\Gamma(k+a)}{\Gamma(k+1)}.
\end{aligned}$$

Based on the preceding equations, $\mathcal{L}_{k+1} - \mathcal{L}_k \approx \mathcal{R}_{k+1} - \mathcal{R}_k$, we have (we denote the mapping function as $f(k;a)$ for parameter $a$) the following:

$$[f(k+1;a) - 1]^a - [f(k;a) - 1]^a = a[N-1]^a \binom{n-1}{k} \mathcal{B}(a+k, n-k). \tag{53}$$

Then we have the following recurrent formula:

$$\boxed{f(k+1;a) = \left[a[N-1]^a \binom{n-1}{k} \mathcal{B}(a+k, n-k)[f(k;a)-1]^a\right]^{1/a} + 1.} \tag{54}$$

And $f(1)$ is given by $\mathcal{L}_1 = \mathcal{R}_1$:

$$f(1;a) = (N-1)[a\mathcal{B}(a,n)]^{1/a} + 1. \tag{55}$$

## C PROPERTIES OF RECURRENT FUNCTION $f$

In the following, we enumerate a list of interesting properties of this recurrent formula of $f$ based on Beta distribution.

Lemma C.1 (Location of Last Point). *For any $a$, all $f(n)$ converge to $N$: $f(n) = N$.*

Proof. We note that

$$\sum_{k=0}^{n-1} \binom{n-1}{k} \mathcal{B}(a+k, n-k) = \int_0^1 \sum_{k=0}^{n-1} \binom{n-1}{k} t^{a+k-1}(1-t)^{n-k-1} dt$$
$$= \int_0^1 t^{a-1} \left[\sum_{k=0}^{n-1} \binom{n-1}{k} t^k (1-t)^{n-k-1}\right] dt = \int_0^1 t^{a-1} = \frac{1}{a}.$$

Adding up Equation (53) from $k = 1$ to $n-1$, we have.

$$\frac{[f(n)-1]^a - [f(1)-1]^a}{a[N-1]^a} = \sum_{k=1}^{n-1} \binom{n-1}{k} \mathcal{B}(a+k, n-k) = \frac{1}{a} - \binom{n-1}{0} \mathcal{B}(a,n), \text{ and then we have } f(n) = N.$$

□

***Uniform Distribution and Linear Map.*** When the parameter $a = 1$, the Beta distribution degenerates to the uniform distribution. From Equations (25) and (24), we have another simple linear map:

$$f(k; a=1) = k\frac{N-1}{n} + 1. \tag{56}$$

Even though the user rank distribution is quite different from the uniform distribution, we found that this formula provides a reasonable approximation for the mapping function and, generally, better than the Naive formula Equation (18). More interestingly, we found that when $a$ ranges from 0 to 1 (as they express an exponential-like distribution), they actually are quite close to this linear formula.

***Approximately Linear.*** When we take a close look at the $f(k;a)$ sequences $(f(1;a), f(2;a), \ldots f(k;a)$ for different parameters $a$ from 0 to 1,[1] we find that when $k$ is large, $f(k;a)$ all gets very close to $f(k;1)$ (the linear map function for the uniform distribution). Figure 5 shows the relative difference of all $f(k;a)$ sequences for $a = 0.2, 0.6, 0.8$ with respect to $a = 1$ (i.e., $[f(k;a) - f(k;a=1)]/f(k;a=1)$). Basically, they all converge quickly to $f(k;a=1)$ as $k$ increases.

[1]This also holds when $a > 1$, but since the *Recall* (a.k.a. the user rank distribution) is typically very different from these settings, we do not discuss them here.

To observe this, let us take a look at their $f(K)$ locations when $k$ is getting large. To simplify our discussion, let $g(k) = f(K) - 1$, and then we have

$$[g(k+1)]^a - [g(k)]^a = a(N-1)^a \frac{\Gamma(n)}{\Gamma(n+a)} \frac{\Gamma(k+a)}{\Gamma(k+1)}$$

$$\left[\frac{g(k+1)}{g(k)}\right]^a = 1 + \left[\frac{(N-1)k}{g(k)n}\right]^a \frac{a}{k},$$

and when $n$ and $k$ are large, $\lim_{n\to\infty} \frac{\Gamma(n)}{\Gamma(n+a)} = \frac{1}{n^a}$

$$\left[\frac{g(k+1)}{g(k)}\right] = \left(1 + \left[\frac{(N-1)k}{g(k)n}\right]^a \frac{a}{k}\right)^{1/a} \approx 1 + \left[\frac{(N-1)k}{g(k)n}\right]^a \frac{1}{k}.$$

When $g(k) = (N-1)\frac{k}{n}$, the preceding equation holds $\frac{g(k+1)}{g(k)} = 1 + \frac{1}{k}$, and this suggests that they are all quite similar to the linear map $f(k; a = 1)$ for the uniform distribution.

By looking at the difference $f(k;a) - f(k-1;a)$, we notice we will get very close to the constant $\frac{N-1}{n} = f(k;1) - f(k-1;1)$ even when $k$ is small. To verify this, let

$$y_{k+1} = [f(k+1;a) - 1]^a - [f(k;a) - 1]^a = a[N-1]^a \frac{\Gamma(n)}{\Gamma(n+a)} \frac{\Gamma(k+a)}{\Gamma(k+1)}.$$

Then we immediately observe the following:

$$\frac{y_{k+1}}{y_k} = \frac{a[N-1]^a \frac{\Gamma(n)}{\Gamma(n+a)} \frac{\Gamma(k+a)}{\Gamma(k+1)}}{a[N-1]^a \frac{\Gamma(n)}{\Gamma(n+a)} \frac{\Gamma(k-1+a)}{\Gamma(k)}} = 1 + \frac{a-1}{k}.$$

Thus, after only a few iterations for $f(k;a)$, we have found that their (powered) difference will get close to being a constant.

## D FIX-SIZE-SAMPLING ESTIMATION: THE EM ALGORITHM

In this section, we give the details of the EM algorithm for the MLE estimator. Recalling Equation (32), the weighted log-likelihood function is as follows:

$$\log \mathcal{L} = \sum_{u=1}^{M} w_u \cdot \log \sum_{k=1}^{N} p(x_u, z_{uk} | \theta_k).$$

*E-step.*

$$Q(\boldsymbol{\pi}, \boldsymbol{\pi}^{old}) = \sum_{u=1}^{M} w_u \sum_{k=1}^{N} \gamma(z_{uk}) \log p(x_u, z_{uk} | \theta_k),$$

where

$$\gamma(z_{uk}) = p(z_{uk} | x_u, \boldsymbol{\pi}^{old}) = \frac{\pi_k^{old} p(x_u | \theta_k)}{\sum_{j=1}^{N} \pi_j^{old} p(x_u | \theta_j)}.$$

*M-step.*

$$Q'(\boldsymbol{\pi}, \boldsymbol{\pi}^{old}) = Q(\boldsymbol{\pi}, \boldsymbol{\pi}^{old}) + \lambda \left(1 - \sum_{k=1}^{N} \pi_k\right) \tag{57}$$

$$\lambda = \sum_{k=1}^{N} \sum_{u=1}^{M} w_u \cdot \gamma(z_{uk}) = \sum_{u=1}^{M} w_u \tag{58}$$

$$\pi_k^{new} = \frac{\sum_{u=1}^{M} w_u \cdot \gamma(z_{uk})}{\sum_{u=1}^{M} w_u}$$

## E LINEAR COMBINATION OF COEFFICIENTS

Considering $X = (X_1, \ldots, X_n)$ is the random variables of a sample $M$ times multinomial distribution with $n$ cells probabilities $(\theta_1, \ldots, \theta_n)$. We have $\frac{X_i}{M} \to \theta_i$, when $M \to \infty$.

$$\begin{aligned} &\mathbb{E}[X_i] = M\theta_i \quad Var[X_i] = M\theta_i(1-\theta_i) \\ &Cov(X_i, X_j) = -M\theta_i\theta_j \end{aligned} \tag{59}$$

Considering the new random variable deriving from the linear combination: $\mathcal{A} = \sum_{i=1}^{n} w_i X_i$, where the $w_i$ are the constant coefficients.

$$\begin{aligned} \mathbb{E}[\mathcal{A}] &= M \cdot \sum_{i=1}^{n} w_i\theta_i \\ Var[\mathcal{A}] = \mathbb{E}[\mathcal{A}^2] - (\mathbb{E}[\mathcal{A}])^2 &= \sum_{i}^{n} w_i^2[M\theta_i - M\theta_i^2] - 2\sum_{i\neq j} w_i w_j[M\theta_i\theta_j] \\ &= M\sum_{i}^{n} w_i^2\theta_i - M\cdot\Big(\sum_{i}^{n} w_i^2\theta_i^2 + 2\sum_{i\neq j} w_i w_j\theta_i\theta_j\Big) \\ &= M\cdot\Big(\sum_{i}^{n} w_i^2\theta_i - \big(\sum_{i}^{n} w_i\theta_i\big)^2\Big) \end{aligned}$$

## F REWRITING OF $L_2$

$$\begin{aligned} \mathcal{L}_2 &= \sum_{R=1}^{N} P(R)\cdot\mathbb{E}\Big[\sum_{r=1}^{n}\tilde{P}(r|R)\widehat{\mathcal{M}}(r) - \sum_{r=1}^{n} P(r|R)\widehat{\mathcal{M}}(r)\Big]^2 = \sum_{R=1}^{N} P(R)\cdot\mathbb{E}\Big[\sum_{r=1}^{n}\frac{X_r}{M\cdot P(R)}\widehat{\mathcal{M}}(r) - \sum_{r=1}^{n}\frac{\mathbb{E}[X_r]}{M\cdot P(R)}\widehat{\mathcal{M}}(r)\Big]^2 \\ &= \sum_{R=1}^{N}\frac{1}{M^2\cdot P(R)}\cdot\mathbb{E}\Big[\sum_{r=1}^{n} X_r\widehat{\mathcal{M}}(r) - \sum_{r=1}^{n}\mathbb{E}[X_r]\widehat{\mathcal{M}}(r)\Big]^2 = \sum_{R=1}^{N}\frac{1}{M}\cdot\Big(\sum_{r}^{n}\widehat{\mathcal{M}}^2(r)P(r|R) - \big(\sum_{r}^{n}\widehat{\mathcal{M}}(r)P(r|R)\big)^2\Big) \\ &= \sum_{R=1}^{N}\frac{1}{M}Var(\widehat{\mathcal{M}}(r)|R) \end{aligned}$$

## G REWRITE OF $L$

$$\begin{aligned} \mathcal{L}_1 &= \sum_{R=1}^{N} P(R)\Big(\sum_{r=1}^{n} P(r|R)\widehat{\mathcal{F}}(r) - \mathcal{F}(R)\Big)^2 = ||\sqrt{D}A\mathbf{x} - \sqrt{D}\mathbf{b}||_F^2 \\ \mathcal{L}_2 &= \sum_{R=1}^{N} P(R)\cdot\mathbb{E}\Big[\sum_{r=1}^{n}\tilde{P}(r|R)\widehat{\mathcal{F}}(r) - \sum_{r=1}^{n} P(r|R)\widehat{\mathcal{F}}(r)\Big]^2 = \sum_{R=1}^{N} P(R)\cdot\mathbb{E}\Big[\sum_{r=1}^{n}\frac{X_r}{M\cdot P(R)}\widehat{\mathcal{F}}(r) - \sum_{r=1}^{n}\frac{\mathbb{E}[X_r]}{M\cdot P(R)}\widehat{\mathcal{F}}(r)\Big]^2 \\ &= \sum_{R=1}^{N}\frac{1}{M^2\cdot P(R)}\cdot\mathbb{E}\Big[\sum_{r=1}^{n} X_r\widehat{\mathcal{F}}(r) - \sum_{r=1}^{n}\mathbb{E}[X_r]\widehat{\mathcal{F}}(r)\Big]^2 = \sum_{R=1}^{N}\frac{1}{M}\cdot\Big(\sum_{r}^{n}\widehat{\mathcal{F}}^2(r)P(r|R) - \big(\sum_{r}^{n}\widehat{\mathcal{F}}(r)P(r|R)\big)^2\Big) \\ &= \frac{1}{M}\mathbf{x}^T\Lambda_1\mathbf{x} - \frac{1}{M}||A\mathbf{x}||_F^2 = \frac{1}{M}||\sqrt{\Lambda_1}\mathbf{x}||_F^2 - \frac{1}{M}||A\mathbf{x}||_F^2 \end{aligned}$$

## H ADAPTIVE EM STEPS

***E-step.***

$$\begin{aligned}\log \mathcal{L} &= \sum_{u=1}^{M} \log \sum_{k=1}^{N} P(r_u, R_{uk}; \boldsymbol{\pi}) = \sum_{u=1}^{M} \log \sum_{k=1}^{N} \phi(R_{uk}) \cdot \frac{P(r_u, R_{uk}; \boldsymbol{\pi})}{\phi(R_{uk})} \\ &\geq \sum_{u=1}^{M} \sum_{k=1}^{N} \phi(R_{uk}) \cdot \log P(r_u, R_{uk}; \boldsymbol{\pi}) + constant \\ &\triangleq \sum_{u=1}^{M} Q_u(\boldsymbol{\pi}, \boldsymbol{\pi}^{old}) = Q(\boldsymbol{\pi}, \boldsymbol{\pi}^{old}),\end{aligned} \tag{60}$$

where

$$\phi(R_{uk}) = P(R_u = k | r_u; \boldsymbol{\pi}^{old}) = \frac{\pi_k^{old} \cdot P(r_u | R_u = k; n_u)}{\sum_{j=1}^{N} \pi_j^{old} \cdot P(r_u | R_u = j; n_u)}, \tag{61}$$

where $\phi$ is the posterior, $\boldsymbol{\pi}^{old}$ is the known probability distribution, and $\boldsymbol{\pi}$ is parameter.

***M-step.*** Derived from the Lagrange maximization procedure:

$$\pi_k^{new} = \frac{1}{M} \sum_{u=1}^{M} \phi(R_{uk}). \tag{62}$$

## REFERENCES


[1] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press. https://doi.org/10.1017/CBO9780511804441

[2] G. Casella and R. L. Berger. 2002. *Statistical Inference*. Thomson Learning.

[3] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience.

[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*. 1927–1932.

[5] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. 39–46.

[6] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2019. A troubling analysis of reproducibility and progress in recommender systems research. *arXiv:1911.07698* (2019).

[7] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. 2021. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys '21)*.

[8] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

[9] Mukund Deshpande and George Karypis. 2004. Item-based top-$N$ recommendation algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177.

[10] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*. 515–524.

[11] Ali Mamdouh Elkahky, Y. Song, and X. He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. 278–288.

[12] Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef. 2020. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences* 10, 21 (2020), 7748.

[13] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline evaluation to make decisions about PlaylistRecommendation algorithms. In *Proceedings*

*of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, New York, NY, 420–428. https://doi.org/10.1145/3289600.3291027

[14] Asela Gunawardana and Guy Shani. 2009. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research* 10, 12 (2009), 2935–2962.

[15] U. Gupta, S. Hsia, V. Saraph, X. Wang, B. Reagen, G. Wei, H. S. Lee, D. Brooks, and C. Wu. 2020. DeepRecSys: A system for optimizing end-to-end at-scale neural recommendation inference. In *Proceedings of the 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA '20)*. 982–995.

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 22nd International Conference on World Wide Web (WWW '17)*.

[17] Binbin Hu, C. Shi, W. X. Zhao, and P. S. Yu. 2018. Leveraging meta-path based context for top-*N* recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '18)*. 1531–1540.

[18] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 9th IEEE International Conference on Data Mining (ICDM '08)*.

[19] Ruoming Jin, Dong Li, Jing Gao, Zhi Liu, Li Chen, and Yang Zhou. 2021. Towards a better understanding of linear models for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*. 776–785.

[20] Ruoming Jin, Dong Li, Benjamin Mudrak, Jing Gao, and Zhi Liu. 2021. On estimating recommendation evaluation metrics under sampling. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 5 (May 2021), 4147–4154.

[21] Ruoyan Kong, Charles Chuankai Zhang, Ruixuan Sun, Vishnu Chhabra, Tanushsrisai Nadimpalli, and Joseph A. Konstan. 2022. Multi-objective personalization in multi-stakeholder organizational bulk e-mail: A field experiment. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), Article 528, 27 pages.

[22] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. 426–434.

[23] Walid Krichene, N. Mayoraz, S. Rendle, L. Zhang, X. Yi, L. Hong, Ed H. Chi, and J. R. Anderson. 2019. Efficient training on very large corpora via Gramian estimation. In *Proceedings of the International Conference on Learning Representations (ICLR '19)*.

[24] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '20)*. ACM, New York, NY, 1748–1757.

[25] Erich L. Lehmann and George Casella. 2006. *Theory of Point Estimation*. Springer Science & Business Media.

[26] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. 2020. On sampling top-K recommendation evaluation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '20)*.

[27] Dong Li, Ruoming Jin, Zhenming Liu, Bin Ren, Jing Gao, and Zhi Liu. 2023. Towards reliable item sampling for recommendation evaluation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI '23)*.

[28] Wentian Li, Pedro Miramontes, and Cocho Germinal. 2010. Fitting ranked linguistic data with two-parameter functions. *Entropy* 12, 7 (2010), 1743–1764.

[29] Dawen Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web (WWW '18)*. 689–698.

[30] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and Philip S. Yu. 2020. Basket recommendation with multi-intent translation graph neural network. In *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data '20)*. 728–737.

[31] K. Deergha Rao. 2018. *Signals and Systems*. Springer International Publishing.

[32] Steffen Rendle. 2019. Evaluation metrics for item recommendation under sampling. *arXiv preprint arXiv:1912.02263* (2019).

[33] Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *Proceedings of the World Wide Web Conference (WWW '19)*. 3251–3257.

[34] Harald Steck, Linas Baltrunas, Ehtsham Elahi, Dawen Liang, Yves Raimond, and Justin Basilico. 2021. Deep learning for recommender systems: A Netflix case study. *AI Magazine* 42, 3 (Nov. 2021), 7–18. https://doi.org/10.1609/aimag.v42i3.18140

[35] Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. 2021. Quality metrics in recommender systems: Do we calculate metrics consistently? In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys '21)*. 708–713.

[36] Xiang Wang, D. Wang, C. Xu, X. He, Y. Cao, and T. Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI '19)*.

[37] 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 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*. 664–672.

[38] Longqi Yang, Y. Cui, Y. Xuan, C. Wang, S. J. Belongie, and D. Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommended Systems (RecSys '18)*. 279–287.

[39] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the Conference on Information and Knowledge Management (CIKM '21)*.

Received 1 December 2022; revised 13 September 2023; accepted 2 October 2023