On Sampled Metrics for Item Recommendation

Walid Krichene
walidk@google.com
Google Research
Mountain View, California

Steffen Rendle
srendle@google.com
Google Research
Mountain View, California

ABSTRACT

The task of item recommendation requires ranking a large catalogue of items given a context. Item recommendation algorithms are evaluated using ranking metrics that depend on the positions of relevant items. To speed up the computation of metrics, recent work often uses sampled metrics where only a smaller set of random items and the relevant items are ranked. This paper investigates sampled metrics in more detail and shows that they are inconsistent with their exact version, in the sense that they do not persist relative statements, e.g., *recommender A is better than B*, not even in expectation. Moreover, the smaller the sampling size, the less difference there is between metrics, and for very small sampling size, all metrics collapse to the AUC metric. We show that it is possible to improve the quality of the sampled metrics by applying a correction, obtained by minimizing different criteria such as bias or mean squared error. We conclude with an empirical evaluation of the naive sampled metrics and their corrected variants. To summarize, our work suggests that sampling should be avoided for metric calculation, however if an experimental study needs to sample, the proposed corrections can improve the quality of the estimate.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Evaluation of retrieval results*; • **Computing methodologies** → *Ranking*.

KEYWORDS

Item Recommendation; Evaluation; Metrics; Sampled Metric

ACM Reference Format:
Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394486.3403226

1 INTRODUCTION

Over recent years, item recommendation from implicit feedback has received a lot of attention from the recommender system research community. At its core, item recommendation is a retrieval task, where given a context, a catalogue of items should be ranked and the top scoring ones are shown to the user. Usually the catalogue of items to retrieve from is large: tens of thousands in academic studies and often many millions in industrial applications. Finding matching items from this large pool is challenging as the user will usually only explore a few of the highest ranked ones. For evaluating recommender algorithms, usually sharp metrics such as precision or recall over the few highest scoring items (e.g., top 10) are chosen. Another popular class are smooth metrics such as average precision or normalized discounted cumulative gain (NDCG) which place a strong emphasis on the top ranked items.

Recently, it has become common in research papers to speed up evaluation by sampling a small set of irrelevant items and ranking the relevant documents only among this smaller set [7, 9, 10, 12, 15–17]. Sampling of negatives is commonly used during training of large models [4, 13], and several works have studied the implications of sampling as well as various methods to improve it [5, 6], see [18] for a comparative study of sampling methods. However, to the best of our knowledge, the implications of sampling during *evaluation* have not been explored. In this work, the consequences of this approach are studied. In particular, it is shown that findings from sampled metrics (even in expectation) can be inconsistent with exact metrics. This means that if a recommender A outperforms a recommender B on the sampled metric, it does not imply that A has a better metric than B when the metric is computed exactly. This is even a problem in expectation; i.e., with unlimited repetitions of the measurement. Moreover, a sampled metric has different characteristics than its exact counterpart. In general, the smaller the sampling size, the less differences there are between different metrics, and in the small sample limit, all metrics collapse to the area under the ROC curve, which discounts positions linearly. This is particularly problematic because many ranking metrics are designed to focus on the top positions.

As we will show, the sampled metrics can be viewed as high-bias, low-variance estimators of the exact metrics. Their low variance can be particularly misleading if one does not recognize that they are biased, as repeated measurements may indicate a low variance, and yet no meaningful conclusion can be drawn because the bias is *recommender-dependent*, i.e. the value of the bias depends on the recommender algorithm being evaluated. We also show that this issue can be alleviated if one applies a point-wise correction to the sampled metric, by minimizing criteria that trade-off bias and variance. Empirical performance of the sampled metrics and their corrections is illustrated on a movie recommendation problem.

This analysis suggests that if a study is really interested in metrics that emphasize the top ranked items, sampling candidates should be avoided for the purposes of evaluation, and if the size of the problem is such that sampling is necessary, corrected metrics can provide a more accurate evaluation. Lastly, if sampling is used, the reader should be aware that the reported metric has different characteristics than its name implies.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7998-4/20/08.
https://doi.org/10.1145/3394486.3403226

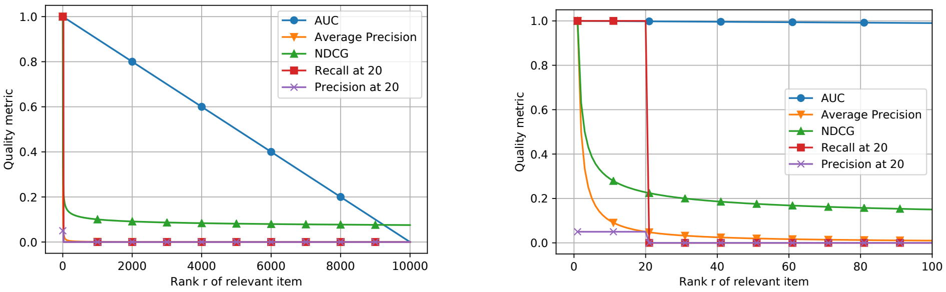

Figure 1: Visualization of metric vs. predicted rank for $n = 10,000$. The left side shows the metrics over the whole set of $10,000$ items. The right side zooms onto the contributions of the top 100 ranks. All metrics besides AUC are top heavy and almost completely ignore the tail. This is usually a desirable property for evaluating ranking because users are unlikely to explore items further down the result list.

2 EVALUATING ITEM RECOMMENDATION

This section starts by formalizing the most common evaluation scheme for item recommendation. Let there be a pool of n items to recommend from. For a given instance[1] $\mathbf{x}$, a recommendation algorithm, A, returns a ranked list of the n items. In an evaluation, the positions, $R(A, \mathbf{x}) \subseteq \{1, \dots, n\}$, of the withheld relevant items within this ranking are computed – R will also be referred to as the *predicted ranks*. For example, $R(A, \mathbf{x}) = \{3, 5\}$ means for an instance $\mathbf{x}$ recommender A ranked two relevant items at positions 3 and 5. Then, a metric M is used to translate the positions into a single number measuring the quality of the ranking. This process is repeated for a set of instances, $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$, and an average metric is reported:

$$\frac{1}{|D|} \sum_{\mathbf{x} \in D} M(R(A, \mathbf{x})). \tag{1}$$

This problem definition assumes that in the ground truth, all relevant items are equally preferred by the user, i.e., that the relevant items are a *set*. This is the most commonly used evaluation scheme in recommender systems. In more complex cases, the ground truth includes preferences among the relevant items. For example, the ground truth can be a ranked list or weighted set. Our work shows issues with sampling in the simpler setup, which implies that the issues carry over to the more complex case.

3 METRICS

This section recalls commonly used metrics for measuring the quality of a ranking. For convenience, the arguments, $A, \mathbf{x}$, from $R(A, \mathbf{x})$ are omitted whenever the particular recommender, A, or instance, $\mathbf{x}$, is clear from context. Instead, the shorter form R is used.

Area under the ROC curve (AUC) measures the likelihood that a random relevant item is ranked higher than a random irrelevant item.

$$\begin{aligned}\mathrm{AUC}(R)_n &= \frac{1}{|R|(n-|R|)} \sum_{r \in R} \sum_{r' \in (\{1,\dots,n\} \setminus R)} \delta(r < r') \\ &= \frac{n - \frac{|R|-1}{2} - \frac{1}{|R|} \sum_{r \in R} r}{n - |R|},\end{aligned} \tag{2}$$

with the indicator function $\delta(b) = 1$ if b is true and 0 otherwise. Precision at position k measures the fraction of relevant items among the top k predicted items:

$$\mathrm{Prec}(R)_k = \frac{|\{r \in R : r \leq k\}|}{k}. \tag{3}$$

Recall at position k measures the fraction of all relevant items that were recovered in the top k:

$$\mathrm{Recall}(R)_k = \frac{|\{r \in R : r \leq k\}|}{|R|}. \tag{4}$$

Average Precision at k measures the precision at all ranks that hold a relevant item:

$$\mathrm{AP}(R)_k = \frac{1}{\min(|R|, k)} \sum_{i=1}^{k} \delta(i \in R) \mathrm{Prec}(R)_i. \tag{5}$$

Normalized discounted cumulative gain (NDCG) at k places an inverse log reward on all positions that hold a relevant item:

$$\mathrm{NDCG}(R)_k = \frac{1}{\sum_{i=1}^{\min(|R|,k)} \frac{1}{\log_2(i+1)}} \sum_{i=1}^{k} \delta(i \in R) \frac{1}{\log_2(i+1)}. \tag{6}$$

3.1 Simplified Metrics

The remainder of the paper analyzes these metrics for $|R| = 1$, i.e., there exists exactly one relevant item which is ranked at position r. This will simplify the analysis and give a better understanding of the differences between these metrics. The metrics of the previous section simplify to the following:

$$\mathrm{AUC}(r)_n = \frac{n-r}{n-1}, \tag{7}$$

$$\mathrm{Prec}(r)_k = \delta(r \leq k) \frac{1}{k}, \tag{8}$$

$$\mathrm{Recall}(r)_k = \delta(r \leq k), \tag{9}$$

$$\mathrm{AP}(r)_k = \delta(r \leq k) \frac{1}{r}, \tag{10}$$

$$\mathrm{NDCG}(r)_k = \delta(r \leq k) \frac{1}{\log_2(r+1)}. \tag{11}$$

[1] E.g., a user, context, or query.

	Predicted Ranks	AUC	AP	NDCG	Recall@10
A	100, 100, 100, 100, 100	**0.990**	0.010	0.150	0.000
B	40, 40, 8437, 9266, 4482	0.555	0.010	0.122	0.000
C	212, 2, 743, 5342, 1548	0.843	**0.101**	**0.208**	**0.200**

Table 1: Toy example of evaluating three recommenders A, B and C on five instances.

	Predicted Ranks	AUC	AP	NDCG	Recall@10
A	100, 100, 100, 100, 100	**0.990**±0.004	**0.630**±0.129	**0.724**±0.097	**1.000**±0.000
B	40, 40, 8437, 9266, 4482	0.555±0.014	0.336±0.073	0.444±0.054	0.400±0.000
C	212, 2, 743, 5342, 1548	0.843±0.014	0.325±0.050	0.460±0.039	0.567±0.092

Table 2: Sampled evaluation for the recommenders from Table 1. On sampled metrics, the relative ordering of A, B, C is not preserved, except for AUC.

For metrics such as Average Precision and NDCG, it makes sense to also define their untruncated counterpart, e.g., for $k = n$:

$$\mathrm{AP}(r) = \frac{1}{r}, \tag{12}$$

$$\mathrm{NDCG}(r) = \frac{1}{\log_2(r+1)}. \tag{13}$$

Some other popular metrics can be reduced to these definitions: For $|R| = 1$, *Reciprocal Rank* is equivalent to Average Precision, *Hit Ratio* is equivalent to Recall and *Accuracy* is equivalent to Recall at 1, and Precision at 1.

Figure 1 visualizes how the different ranking metrics trade-off the position vs. quality score. Average precision has the sharpest score decay, e.g., rank 1 is twice as valuable as rank 2, whereas for NDCG, rank 1 is 1.58 more valuable than rank 2. The least position-aware metric is AUC which places a linear decay on the rank. E.g., pushing an item from position 101 to 100 is as valuable as pushing an item from position 2 to 1.

3.2 Example

This section concludes with a short example that will be used throughout this work. Let there be three recommenders A, B, C and a set of $n = 10,000$ items. Each recommender is evaluated on five instances (i.e., $|D| = 5$) with one relevant item each. For each instance, each recommender creates a ranking and the position at which the relevant item appears is recorded. Assume that recommender C manages to rank the relevant item in one of the evaluation instances on position 2, besides this it never achieves a good rank for the other four instances. Assume recommender B ranks relevant items in two evaluation instances at position 40. And recommender A is never good nor terrible and the relevant items are ranked at position 100 in each of the five instances. Table 1 shows more details about the predicted ranks and the corresponding evaluation metrics. On AUC, recommender A is the best as it cares about all ranks equally. For top heavy metrics (AP, NDCG and Recall), recommender C scores the highest. This example will be revisited in Section 4.2 when sampled metrics are discussed.

4 SAMPLED METRICS

Ranking all items is expensive when the number of items, n, is large. Recently, it has become common to sample a small set of m irrelevant items, add the relevant items, and compute the metrics only on the ranking generated by this subset [7, 9, 10, 12, 15–17]. It is common to pick the number of sampled irrelevant items, m, in the order of a hundred while the number of items n is much larger, e.g., $m = 100$ samples for datasets with $n = \{4k, 10k, 17k, 140k, 2M\}$ items [7, 9, 15], $m = 50$ samples for $n \in \{2k, 18k, 14k\}$ items [10], or $m = 200$ samples for $n \in \{17k, 450k\}$ items [17]. This section will highlight that this approach is problematic. In particular, results can become inconsistent with the exact metrics.

Let $\tilde{R}$ be the ranks of the relevant items among the union of relevant items and the m randomly sampled irrelevant items. It is important to note that $\tilde{R}$ is a random variable, i.e., it depends on the random sample of irrelevant items. The properties of $\tilde{R}$ will be analyzed in Section 4.3.

4.1 Inconsistency of Sampled Metrics

A central goal of evaluation metrics is to make comparisons between recommenders, such as, *recommender A has a higher value than B on metric M*. When comparing recommenders among sampled metrics, we would hope that at least the relative order is preserved in expectation. This property can be formalized as follows.

Definition 1 (Consistency). *Let the evaluation data D be fixed. A metric M is consistent under sampling if the relative order of any two recommenders A and B is preserved in expectation. That is, for all A, B,*

$$\frac{1}{|D|}\sum_{\mathbf{x}\in D} M(R(A,\mathbf{x})) > \frac{1}{|D|}\sum_{\mathbf{x}\in D} M(R(B,\mathbf{x}))$$
$$\Longleftrightarrow E\left[\frac{1}{|D|}\sum_{\mathbf{x}\in D} M(\tilde{R}(A,\mathbf{x}))\right] > E\left[\frac{1}{|D|}\sum_{\mathbf{x}\in D} M(\tilde{R}(B,\mathbf{x}))\right]. \tag{14}$$

If a metric is inconsistent, then measuring M on a subsample is not a good indicator of the true performance of M.

4.2 Example

Now, the example from Section 3.2 is revisited and the same measures are computed using sampling. Specifically, $m = 99$ random irrelevant items are sampled, the position $\tilde{r}$ of the relevant item among this sampled subset is found, and then the metrics are computed for the rank $\tilde{r}$ within the subsample. This procedure with a comparable sample size is commonly used in recent work [7, 9, 10, 15, 17].

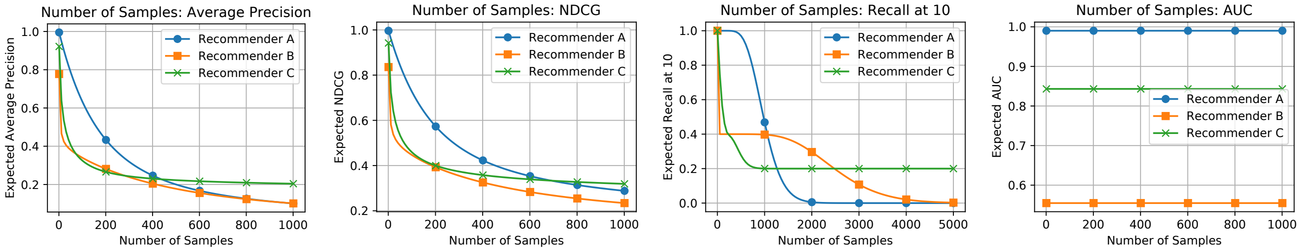

Figure 2: Expected sampling metrics for the running example (Section 3.2 and 4.2) while increasing the number of samples. For Average Precision, NDCG and Recall, even the relative order of recommender performance changes with the number of samples. That means, conclusions drawn from a subsample are not consistent with the true performance of the recommender.

Table 2 shows the sampled metrics for the example from Section 3.2. As this is a random process, for better understanding of its outcome, here it is repeated 1000 times and the average and standard deviation is computed[2].

Compared to the exact metrics in Table 1, even the relative ordering of metrics completely changed. On the exact metrics, C is clearly the best with a 10x higher average precision than B and A. But it has the lowest average precision when sampled measurements are used. A and B perform the same on the exact metrics, but A has a 2x better average precision on the sampled metrics. Sampled average precision does not give any indication of the true ordering among the methods. Similarly, sampled NDCG and sampled Recall at 10 do not agree with the exact metrics. Only AUC is consistent between sampled and exact computation. The other metrics are inconsistent.

Figure 2 shows the same study as in the previous table, as we vary the number of samples, m. The relative ordering of recommenders changes with an increasing sample size. For example, for average precision, depending on the number of samples, any conclusion could be drawn: A better than C better than B (for sample size < 50), A better than B better than C (for sample size ≈ 200), C better than A better than B (for sample size ≈ 500), and finally C better than A equal B (for large sample sizes). This example shows that the bias of sampled average precision is recommender dependent and sample-size dependent. This is why the relative ordering of recommenders changes as we change the sample size. Similar observations can be made for NDCG. Recall is even more sensitive to the sample size, and it takes about $m = 5,000$ samples out of $n = 10,000$ items for the metric to become consistent. Only AUC is consistent for all m, and the expected metric is independent of sample size.

4.3 Rank Distribution under Sampling

This section takes a closer look at the sampling process and derives the distribution of ranks, $\tilde{R}$ and the expected metrics. For simplicity, the analysis is restricted to rankings with exactly one relevant item, i.e., $|\tilde{R}| = 1$, so we can use the simplified metrics from Section 3.1. Let r denote the true rank of the unique relevant item, and $\tilde{r}$ denote its measured rank on the sample.

When an irrelevant item is sampled uniformly, it can either rank higher or lower than the relevant item. If the number of all items is n, then the probability that the sampled item j is ranked above r is:

$$p(j < r) = \frac{r-1}{n-1}. \tag{15}$$

For example, if r is at position 1, the likelihood of a random irrelevant being ranked higher is 0. If $r = n$, then the likelihood is 1. Note that the pool of all possible sampled items excludes the truly relevant item and thus has size $n - 1$.

Repeating the sampling procedure m times with replacement and counting how often an item is ranked higher, corresponds to a Binomial distribution. In other words, the rank $\tilde{r}$ obtained from the sampling process follows $\tilde{r} \sim B\left(m, \frac{r-1}{n-1}\right) + 1$. If there are no successes in getting a higher ranked item, the rank remains 1, if all m samples are successful, the rank is $m + 1$. The expected value of the metrics under this distribution is

$$E[M(\tilde{r})] = \sum_{i=1}^{m+1} p(\tilde{r} = i) M(i). \tag{16}$$

Note that this is implicitly a function of r, m, which appear as parameters of the Binomial distribution. Figure 3 visualizes the expected metrics $E(M(\tilde{r}))$ as we vary r. The figure highlights the weight that the sampled metric assigns to different ranks. Metrics like Average Precision or NDCG are much less top heavy. Even sharp metrics such as recall become smooth. Only AUC remains unchanged. In general, all metrics converge to a linear function in the small sample limit, similar to AUC behavior.

4.4 Expected Metrics

This section analyzes sampled metrics in a more formal way by applying eq. (16) to particular metrics. The discussion focuses on uniform sampling with replacement, i.e., Binomial distributed ranks. Similar results hold for uniform sampling without replacement. In this case, the distribution is hypergeometric, with population size $n - 1$, where a pool of $r - 1$ items can be potential successes. When appropriate, this variation will be discussed as well.

4.4.1 Expected AUC. First, AUC is a linear function of the rank:

$$\mathrm{AUC}_n(r) = \frac{n-r}{n-1} = -\frac{1}{n-1} r + \frac{n}{n-1} = \mathrm{const}_1\, r + \mathrm{const}_2. \tag{17}$$

[2] In a real evaluation, the process would not be repeated because this would contradict the motivation of sampling to reduce computational cost.

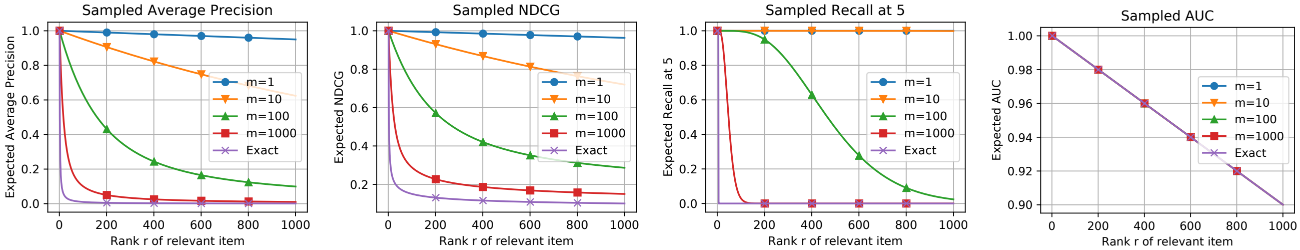

Figure 3: Characteristics of a sampled metric with a varying number of samples. Sampled Average Precision, NDCG and Recall change their characteristics substantially compared to exact computation of the metric. Even large sampling sizes ($m = 1000$ samples of $n = 10000$ items) show large bias. Note this plot zooms into the top 1000 ranks out of $n = 10000$ items.

Thus by linearity of the expectation, and the fact that $\tilde{r}$ follows a Binomial distribution, we have

$$E[\text{AUC}_{m+1}(\tilde{r})] = \text{AUC}_{m+1}(E[\tilde{r}]) = \text{AUC}_{m+1}\left(1 + m\frac{r-1}{n-1}\right)$$
$$= \frac{m+1-1-m\frac{r-1}{n-1}}{m+1-1} = \frac{n-r}{n-1} = \text{AUC}_n(r).$$

That means AUC measurements created by sampling are unbiased estimators of the exact AUC. This result is not surprising because the AUC can alternatively be defined as the expectation that a random relevant item is ranked over a random irrelevant item. Consequently, AUC is a consistent metric under sampling.

This result holds also for any sampling distributions where the expected value of the sampled rank is $1 + m\frac{r-1}{n-1}$. For example, this is also true for sampling from a hypergeometric distribution – i.e., uniform sampling without replacement.

4.4.2 Cut-off metrics. For a cutoff metric such as recall or precision:

$$E[\text{Recall}_k(\tilde{r})] = \sum_{i=1}^{m+1} p(\tilde{r} = i)\text{Recall}_k(i) = \sum_{i=1}^{m+1} p(\tilde{r} = i)\delta(i \le k)$$
$$= \sum_{i=1}^{k} p(\tilde{r} = i) = \text{CDF}\left(k-1; m, \frac{r-1}{n-1}\right). \tag{18}$$

This analysis carries over to any sampling distribution, including the hypergeometric distribution.

4.4.3 Average Precision. For the expected value of sampled average precision, we distinguish two cases. If $r = 1$, then $\tilde{r} = 1$ and the sampled metric is always equal to 1. If $r > 1$, then $p(j < r) > 0$ and

$$E[\text{AP}(\tilde{r})] = \sum_{i=1}^{m+1} p(\tilde{r} = i)\text{AP}(i) = \sum_{i=1}^{m+1} p(\tilde{r} = i)\frac{1}{i}$$
$$= \frac{1-(1-p(j<r))^{m+1}}{p(j<r)\,(m+1)} = \frac{1-\left(\frac{n-r}{n-1}\right)^{m+1}}{(r-1)\,\frac{m+1}{n-1}}. \tag{19}$$

Interestingly, this can be written as:

$$\frac{1-\text{AUC}_n(r)^{m+1}}{r-1}\left(\frac{n-1}{m+1}\right) = \frac{1-\text{AUC}_n(r)^{m+1}}{r-1}\,\text{const}.$$

If $AUC_n(r)^{m+1} \approx 0.0$, this would be $\frac{1}{r-1}$ and would be similar to the unsampled average precision metric. However, as soon as the relevant item is reasonably highly ranked (i.e., AUC is close to 1.0), it takes many samples m for this term to approach 0.

4.4.4 Small Sampling Size. This section investigates the behavior of sampled metrics in the limit, where $m = 1$. In this case, $\tilde{r} \in \{1, 2\}$, and for any metric M and any sampling distribution:

$$E[M(\tilde{r})] = p(\tilde{r} = 1)M(1) + (1 - p(\tilde{r} = 1))M(2).$$

For uniform sampling[3] of items, $p(\tilde{r} = 1)$ is the probability to sample an item that is ranked after r, i.e., is $\frac{n-r}{n-1}$. Now,

$$E[M(\tilde{r})] = \frac{n-r}{n-1}(M(1) - M(2)) + M(2)$$
$$= r\frac{M(2)-M(1)}{n-1} + \frac{n\,M(1)-M(2)}{n-1} = r\,\text{const}_1 + \text{const}_2,$$

which is a linear function of the true rank r, regardless of the metric. If we only care about the ordering produced by two different metrics on a set of rankings (eq. 14), we can ignore const_2. Similarly, for const_1, only the sign matters when comparing two sets of ranking. This sign of $M(2) - M(1)$ depends on how much ranking a relevant item at position 1 is preferred over ranking it at position 2. For metrics that cannot distinguish between the first and second position, such as precision and recall at $k \ge 2$, the sampled metric is always constant and not useful at all. For any reasonable metric, const_1 should be negative, i.e., ranking at position 1 gives a higher metric than position 2. To summarize, for $m = 1$ all metrics give the same qualitative result in expectation. There is no reason to choose one metric over the other if we are only interested in relative statements such as "metric of A is higher than metric of B". Furthermore, the qualitative result with $m = 1$ coincides with exhaustive AUC since (i) all sampled metrics, including sampled AUC, are indistinguishable for $m = 1$ as shown in this section, and (ii) sampled AUC is consistent with exhaustive AUC as shown in Section 4.4.1.

The discussion above shows that it does not make sense to choose different metrics for $m = 1$; any sensible metric gives the same qualitative statement. A similar observation can be found in Figure 3 and 2 where all metrics behave similarly for small samples sizes.

5 CORRECTED METRICS

So far, we have shown that sampled metrics have different characteristics than the same metric on the full set of items. This section investigates whether we can design a sampled metric $\hat{M}$, a function

[3] Here $m = 1$, so it does not matter whether sampling is with or without replacement.

from $\{1, \ldots, m+1\}$ to $\mathbb{R}$, such that $\hat{M}(\tilde{r})$ provides a good estimate of $M(r)$. We will consider different definitions of what a "good" estimate is.

5.1 Unbiased Estimator of the Rank

Our first approach is motivated by a simple observation. The sampled metrics that are commonly used are obtained by applying the exact metric M to the observed rank $\tilde{r}$, i.e. $\hat{M}(\tilde{r}) = M(\tilde{r})$. But $\tilde{r}$ is a poor estimate of the true rank r, in fact it always under-estimates it. Instead, one can measure the metric not on the observed rank $\tilde{r}$, but on an unbiased estimator of r. Recall from Section 4.3 that $\tilde{r}|r \sim B\left(m, \frac{r-1}{n-1}\right) + 1$. If we let $p := \frac{r-1}{n-1}$, then an unbiased estimator of p is given by $\frac{\tilde{r}-1}{m}$. Thus an unbiased estimator of $r = 1 + (n-1)p$ is given by $\hat{r} := 1 + \frac{(n-1)(\tilde{r}-1)}{m}$. This motivates using the following corrected metric:

$$\hat{M}(\tilde{r}) = M\left(1 + \frac{(n-1)(\tilde{r}-1)}{m}\right). \tag{20}$$

Since the rank estimate is a real number in $[1, n]$, and the original metric M is only defined on natural numbers, we can either round the rank estimate or extend M using e.g. linear interpolation. In our experiments, we round using floor $\lfloor \cdot \rfloor$.

5.2 Minimal Bias Estimator

The first correction used an unbiased estimator of the rank. However, whenever M is nonlinear, $\hat{M}(\tilde{r}) = M(\hat{r})$ is biased in general. A criterion one may seek to optimize is the average bias of $\hat{M}(\tilde{r})$, that is, $\sum_r p(r)(E[\hat{M}(\tilde{r})|r] - M(r))^2$, where $p(r)$ is a prior on the distribution of ranks, if available[4], or the uniform distribution otherwise. Since $\hat{M}$ is a function from $\{1, \ldots, m+1\}$ to $\mathbb{R}$, $\hat{M}$ can equivalently be viewed as a vector in $\mathbb{R}^{m+1}$. Thus we seek to find a vector $\hat{M}$ that minimizes the following problem:

$$\begin{aligned} &\operatorname*{argmin}_{\hat{M} \in \mathbb{R}^{m+1}} \sum_{r=1}^{n} p(r)(E[\hat{M}_{\tilde{r}}|r] - M(r))^2 \qquad (21)\\ = &\operatorname*{argmin}_{\hat{M} \in \mathbb{R}^{m+1}} \sum_{r=1}^{n} p(r)\left(\sum_{\tilde{r}} p(\tilde{r}|r)\hat{M}_{\tilde{r}} - M(r)\right)^2. \end{aligned}$$

This is a least squares problem, and its solution is given by

$$\hat{M} = \left(A^T A\right)^{-1} A^T \mathbf{b}, \tag{22}$$

where

$$\begin{aligned} A \in \mathbb{R}^{n \times m+1}, \quad A_{r,\tilde{r}} &= \sqrt{p(r)}p(\tilde{r}|r),\\ \mathbf{b} \in \mathbb{R}^n, \quad b_r &= \sqrt{p(r)}M(r). \end{aligned} \tag{23}$$

Note that the problem is under-determined when $m + 1 < n$, i.e. in general, one cannot obtain an unbiased estimator for all r. This is consistent with the observation made in Section 4.4.4, that for the limit case $m = 1$, any metric coincides with (an affine transformation of) AUC.

It may also be desirable for the solution $\hat{M}$ to be monotone non-increasing, so that on any given evaluation point, a higher rank $\tilde{r}$ results in a lower estimated metric $\hat{M}_{\tilde{r}}$, although this constraint is not essential when averaging over a large number of evaluation points. The monotonic constraint corresponds to the linear inequalities $\hat{M}_{\tilde{r}+1} \geq \hat{M}_{\tilde{r}}$ for all $\tilde{r}$. In this case, problem (21) becomes an isotonic regression problem [2]. We will refer to this as *Constrained Least Squares* in the experiments.

5.3 Bias-Variance Trade-off

One potential issue with the minimal bias estimator is that it could have high variance, which we observe numerically in Section 6. In order to alleviate this problem, we can regularize problem (21) by introducing a variance term:

$$\operatorname*{argmin}_{\hat{M} \in \mathbb{R}^{m+1}} \sum_{r=1}^{n} p(r)\left((E[\hat{M}_{\tilde{r}}|r] - M(r))^2 + \gamma \operatorname{Var}[\hat{M}_{\tilde{r}}|r]\right), \tag{24}$$

where γ is a positive constant. This is a regularized least squares problem and its solution is given by:

$$\hat{M} = \left((1.0 - \gamma)A^T A + \gamma \operatorname{diag}(\mathbf{c})\right)^{-1} A^T \mathbf{b}, \tag{25}$$

with A and $\mathbf{b}$ from eq. (23) and $c_{\tilde{r}} = \sum_{r=1}^{n} p(r)p(\tilde{r}|r)$. When $\gamma = 0$, this reduces to problem (21). When $\gamma = 1$, this reduces to the least squares estimator and the solution is

$$\hat{M}_{\tilde{r}} = \frac{\sum_{r=1}^{n} p(\tilde{r}|r)p(r)M(r)}{\sum_{r=1}^{n} p(\tilde{r}|r)p(r)} = \sum_{r=1} p(r|\tilde{r})M(r). \tag{26}$$

In a real study, measurements are aggregated over many evaluation points, which reduces the overall variance, so a lower value $\gamma < 1$ is preferable.

5.4 Example

Figure 4 shows an example of a corrected average precision metric $\hat{\text{AP}}$, for several choices of the parameter γ, and for a uniform prior $p(r)$. The sample size is $m = 100$ and the full item set is $n = 10000$, i.e., a sampling rate of 1%. As can be seen, when no order constraint is applied, lower values of γ give oscillating solutions on the sample (left figure). This is not a problem in aggregate over the full evaluation set (right figure). All corrected sampled metrics are closer, in expectation, to the true metric.

5.5 Effect of the Sample Size and Data Set Size

Increasing the sample size m reduces the bias of the sampled metrics, as seen in Figure 3, as well as the corrected metrics: for example, the solution in (21) has a lower value when optimizing over a higher dimensional vector $\hat{M} \in \mathbb{R}^{m+1}$. Increasing the size of the item set, n, has the opposite effect. Increasing the number of evaluation points, $|D|$, decreases the variance of the average estimates. This mostly benefits the corrected metrics introduced in this section; the uncorrected metrics are high-bias estimators which will have a large error even in the limit of zero variance.

6 EXPERIMENTS

In this section, we study sampled metrics on real recommender algorithms and a real dataset. We investigate: (1) Do recommender algorithms create different ranking distributions, e.g., some are better in the top, some are better overall? (2) Are results from sampled metrics and exact metrics inconsistent, e.g., a given recommender is

[4] Note that the true distribution of ranks $p(r)$ is algorithm dependent and typically unknown. We use a uniform prior in our experiments.

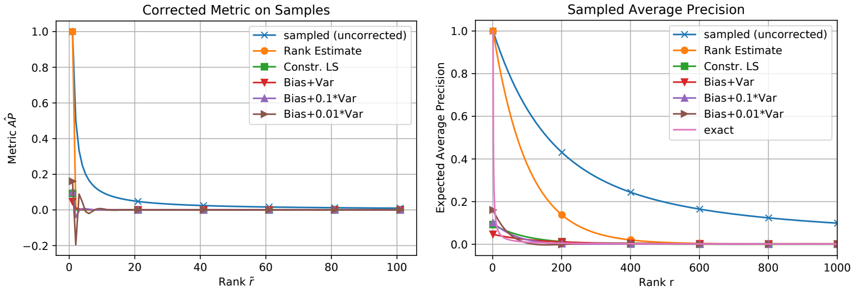

Figure 4: Evaluating the corrected metric $\hat{\text{AP}}$ on a sample of $m = 100$ items (left) is equivalent to measuring the metric on the full item set of $m = 10,000$ (right). Different choices of correction algorithms are plotted.

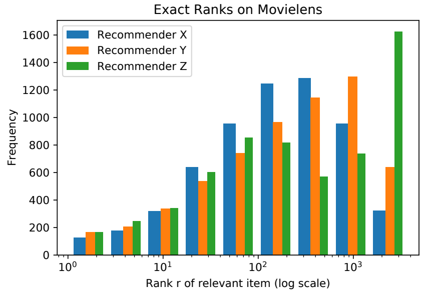

Figure 5: Distribution of predicted ranks for three recommender algorithms on the Movielens 1M dataset.

better on the sampled metric but worse on the true metric? (3) Can corrections help to get more reliable results?

We use an identical experimental setup as [9] – in particular the same dataset (binarized Movielens 1M [8]), split (holdout last event), sampling size ($m = 100$) and metrics (Recall@10=HR@10 and NDCG@10). In addition, we report AP and AUC. We study the behavior of sampled metrics on three popular recommender system algorithms: matrix factorization and two variations of item-based collaborative filtering (see Section 9 in the appendix for details). We want to emphasize that the purpose of our study is not to judge if a particular recommender algorithm is good. The purpose is rather to assess the behavior of metrics and correction methods. To de-emphasize the particular recommender method and hyperparameter choice, we will refer to matrix factorization as 'recommender X', to the two item-based collaborative filtering variations as 'recommender Y' and 'recommender Z'.

6.1 Rank Distributions

For each of the 6040 test users, we rank all items (leaving out the user's training items) and record at which position the withheld relevant item appears. In total we get 6040 ranks. Figure 5 shows the distribution of these ranks. The plot indicates the different characteristics of the three recommenders. Z is the best in the top 10 but has very poor performance at higher ranks as it puts the relevant items of over 1600 users in the worst bucket. X is more balanced and puts only few items at poor ranks; 2310 items are in the top 100 and less than 300 are in the bottom half. Y is in the middle, with a better top 10 performance than X, but tends to put the relevant item at a worse rank overall.

6.2 Sampled Metrics

The leftmost block in Table 3 reports the exact metric and the sampled metric with standard deviation[5]. As expected from the rank distributions, for Recall, NDCG and AP, recommender Z is better than Y better than X on the exact metric. However, on the sampled metric this does not hold. For sampled Recall, the order is reversed and recommender X is much better than Y which is better than Z. All the measures have low standard deviation, so the issue is not that of variance, but is due to the bias in the sampled metrics. Also for NDCG and AP, the worst recommender on the exact metric (X) appears to be the best according to sampled metrics. The relative ordering of the two better recommenders is correct. For AUC, all sampled results are consistent with the exact metrics.

These results indicate that sampled metrics can be inconsistent in real experiments. In particular, if a study would have compared the recommenders only on the sampled metrics, the study would have drawn the wrong conclusion about the performance of the recommender with respect to top heavy metrics such as Recall, NDCG and AP. The worst recommender (X) would have been found to be the best one.

6.3 Corrected Metrics

We finally investigate if correction methods can help. We consider the three correction strategies proposed in Section 5: *rank estimate* (eq. 20), *constrained least squares (CLS)* (eq. 21) with the constraint $\hat{M}_{\tilde{r}} \geq \hat{M}_{\tilde{r}+1}$, and *bias-variance trade-off (BV γ)* with $\gamma \in \{1.0, 0.1, 0.01, 0.001\}$ and a uniform rank distribution $p(r) = 1/n$.

The right block of Table 3 shows the expected metrics under correction. All methods are closer to the exact results than sampling without correction. In particular, CLS and BV with low γ have values close to the exact metric – which indicates a low bias. All identify the order better, e.g., all of them place recommender Z as the best performing method for Recall, NDCG and AP. Some of them (BV with low γ) also get the order of recommenders X and Y right. Figure 6 shows the expected Recall@10 for different choices of the sampling size m. As we can see, the uncorrected metric performs poorly and needs more than $m = 1000$ samples (equivalent to 1/3rd sampling rate) to correctly order recommenders X and Y. The corrected metric using a bias-variance trade-off with $\gamma = 0.1$ already has the correct ordering with less than $m = 60$ samples.

While the corrections seem to be effective in expectation, one also needs to consider the variance of these measurements. Table 4 investigates the bias and variance in more detail. For each sampled metric and each pair of recommenders, we compare the order of the pair over the 100 runs, and count how often the order is correct, i.e. agrees with that of the exact metric. For example, for Recall and "X vs Y" we count in how many of the 100 runs, the metric of recommender X is worse than Y. Table 4 shows that the correction methods are able to resolve most of the mistakes of the uncorrected

[5] We repeated the sampling experiment 100 times to measure the variance.

			Sampled	Sampled with Correction					
	Recommender	Exact	(uncorrected)	Rank Estimate	CLS	BV 1	BV 0.1	BV 0.01	BV 0.001
Recall	X	7.60	66.19±0.25	17.46±0.32	8.52±0.16	4.71±0.07	6.49±0.25	7.18±0.59	7.32±1.17
	Y	8.84	56.51±0.22	17.26±0.28	8.42±0.14	4.60±0.07	6.95±0.21	8.18±0.51	8.54±1.07
	Z	9.42	54.20±0.22	18.67±0.32	9.10±0.15	4.97±0.07	7.44±0.24	8.63±0.59	9.08±1.20
NDCG	X	3.76	39.21±0.20	17.46±0.32	3.99±0.07	2.16±0.04	3.00±0.12	3.34±0.32	3.41±0.71
	Y	4.59	34.82±0.16	17.26±0.28	3.94±0.06	2.12±0.03	3.24±0.10	3.85±0.28	4.03±0.66
	Z	4.79	35.34±0.16	18.67±0.32	4.27±0.07	2.29±0.04	3.46±0.12	4.05±0.32	4.31±0.74
AP	X	3.75	32.55±0.21	18.12±0.31	3.58±0.06	2.44±0.03	3.13±0.09	3.37±0.21	3.42±0.49
	Y	4.32	30.01±0.20	17.81±0.28	3.54±0.06	2.32±0.03	3.19±0.07	3.62±0.18	3.73±0.45
	Z	4.44	30.71±0.21	19.20±0.31	3.82±0.06	2.45±0.03	3.38±0.08	3.79±0.21	3.97±0.51
AUC	X	89.13	89.12±0.04	89.24±0.04	89.12±0.04	88.36±0.04	89.04±0.04	89.11±0.04	89.12±0.04
	Y	85.33	85.33±0.04	85.48±0.04	85.32±0.04	84.63±0.04	85.26±0.04	85.32±0.04	85.32±0.04
	Z	74.73	75.04±0.23	75.24±0.20	75.04±0.21	74.51±0.23	75.02±0.20	75.02±0.24	75.02±0.23

Table 3: Evaluation of three recommenders (X, Y and Z) on the Movielens dataset. Sampled metrics are inconsistent with the exact metrics. Corrected metrics, especially Bias^2+$\gamma*$Variance with $\gamma \leq 0.1$ produce the correct relative ordering in expectation.

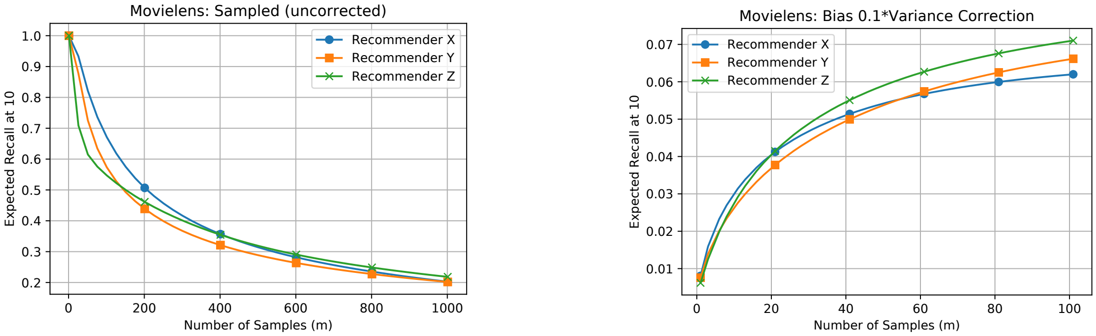

Figure 6: Evaluating recommenders with a varying sample size m. Plots show expected Recall@10 for the uncorrected metric and the metric corrected by Bias^2 + $0.1 *$ Variance. The uncorrected metric needs $m = 1000$ samples to order X and Y correctly in expectation, while for the corrected metric requires only $m = 60$.

		Sampled	Sampled with Correction					
	Measure	(uncorrected)	Rank Estimate	CLS	BV 1	BV 0.1	BV 0.01	BV 0.001
X vs Y	Recall	0	31	31	11	93	91	78
	NDCG	0	31	31	15	93	88	76
	AP	0	24	31	0	68	79	66
	AUC	100	100	100	100	100	100	100
X vs Z	Recall	0	100	100	100	100	95	86
	NDCG	0	100	100	100	100	95	82
	AP	0	100	100	61	99	92	81
	AUC	100	100	100	100	100	100	100
Y vs Z	Recall	0	100	100	100	95	72	68
	NDCG	100	100	100	100	94	70	67
	AP	100	100	100	100	98	75	68
	AUC	100	100	100	100	100	100	100

Table 4: For the 100 repetitions of the experiment in Table 3, how many times the metric for a pair of recommenders show the correct ordering. For example: for Recall and "X vs Y", how often the sampled metric of X was smaller than the sampled metric of Y. In any of the comparisons, a value of 100 indicates the evaluation was always correct, 0 indicates it was always wrong. The exact metric would always score 100.

metrics. For this particular experiment, BV 0.1 seems to be the most effective one, getting the correct order on all but one comparison with >90% chance. The simple 'rank estimate' correction is surprisingly effective and is strictly better than the uncorrected metric in all comparisons of Table 4. It is worth mentioning that under the 'rank estimate' correction, Recall@10 and NDCG@10 are identical. However, it still represents an improvement over the uncorrected metrics which are much more biased and lead to the wrong conclusion. Rank estimate method is trivial to implement (i.e., upscaling the rank before applying the metric). In a study with sampled evaluation, this should be preferred over uncorrected metrics. More complex corrections such as the adjusted bias-variance can get higher gains but are more difficult to implement.

7 SUGGESTIONS

Our results have shown that a sampled metric can be a poor indicator of the true performance of recommender algorithms under this metric. For uncorrected metrics this is mostly due to the large bias introduced by sampling. Using correction methods, this bias can be reduced but at the cost of higher variance. If a study needs to use sampled metrics and is still interested in the true performance of the metrics, we suggest to use a correction method as proposed in this work. In this case it is important to rerun the experiment with different samples (e.g., different random seeds). It is already common, in most evaluations, to repeat an experiment N times – usually by varying the dataset (e.g., N-fold cross validation). In this case, variance is introduced by the differences in the dataset split and potentially by the initialization of the recommender algorithm. In a sampled evaluation, adding a different seed for negative sampling will add another source of variance. That means it may be harder to find "statistically significant" differences between two recommenders. If even under the increased variance a difference is found, then this is a stronger indication that the recommender is truly better under the exact metric. The lower the bias in the corrected metric (e.g., the lower γ), the stronger the indication. While this evaluation is preferable over uncorrected metrics, it is still prone to either not identifying differences (due to variance) or drawing false conclusions because of the bias. This bias can only be eliminated by avoiding sampling altogether.

8 CONCLUSION

This work seeks to bring attention to some issues with sampling of evaluation metrics. It has shown that most metrics are inconsistent under sampling and can lead to false discoveries. Moreover, metrics are usually motivated by applications, e.g., does the top 10 list contain a relevant item? Sampled metrics do not measure the intended quantities – not even in expectation. For this reason, sampling should be avoided as much as possible during evaluation. If an experimental study needs to sample, we propose correction methods that give a better estimate of the true metric, however at the cost of increased variance. Our analysis focused on the case of a single relevant item. The general case may be treated by making the approximation that observed ranks are independent, in which case similar correction methods can be applied. Deriving correction methods without independence is an interesting direction for future research.

ACKNOWLEDGEMENTS

We would like to thank Nicolas Mayoraz and Li Zhang for their helpful comments and suggestions.

REFERENCES

[1] Fabio Aiolli. 2013. Efficient Top-n Recommendation for Very Large Scale Binary Rated Datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems* (Hong Kong, China) *(RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 273–280. https://doi.org/10.1145/2507157.2507189

[2] R.E. Barlow, D.J. Bartholomew, J. M. Bremner, and Brunk H. D. 1972. *Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression.* J. Wiley.

[3] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A Generic Coordinate Descent Framework for Learning from Implicit Feedback. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. 1341–1350. https://doi.org/10.1145/3038912.3052694

[4] Yoshua Bengio and Jean-Sébastien Senecal. 2003. Quick Training of Probabilistic Neural Nets by Importance Sampling. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS 2003, Key West, Florida, USA, January 3-6, 2003.*

[5] Yoshua Bengio and Jean-Sébastien Senecal. 2008. Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model. *IEEE Trans. Neural Networks* 19, 4 (2008), 713–722.

[6] Guy Blanc and Steffen Rendle. 2018. Adaptive Sampled Softmax with Kernel Based Sampling. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 590–599.

[7] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. ACM, New York, NY, USA, 515–524. https://doi.org/10.1145/3209978.3209991

[8] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 173–182. https://doi.org/10.1145/3038912.3052569

[10] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip 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 & Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 1531–1540. https://doi.org/10.1145/3219819.3219965

[11] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08)*. 263–272.

[12] Walid Krichene, Nicolas Mayoraz, Steffen Rendle, Li Zhang, Xinyang Yi, Lichan Hong, Ed Chi, and John Anderson. 2019. Efficient Training on Very Large Corpora via Gramian Estimation. In *International Conference on Learning Representations.*

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (Hong Kong, Hong Kong) *(WWW '01)*. Association for Computing Machinery, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071

[15] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI) (AAAI '19)*. 5329–5336.

[16] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, and Deborah Estrin. 2018. OpenRec: A Modular Framework for Extensible and Adaptable Recommendation Algorithms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. ACM, New York, NY, USA, 664–672. https://doi.org/10.1145/3159652.3159681

[17] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-not-at-random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. ACM, New York, NY, USA, 279–287. https://doi.org/10.1145/3240323.3240355

[18] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. 2017. Selection of Negative Samples for One-class Matrix Factorization. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. 363–371.

9 REPRODUCIBILITY

We provide further details about the evaluation in Section 6 to facilitate reproducibility. There are many different variations of matrix factorization and item-based collaborative filtering. In this section, we clarify the exact model, loss function, and hyper-parameters that we used.

Let U be the set of users, I be the set of items, and $H \subseteq U \times I$ be the training set of items that the user has rated in the past.

Recommender X is a matrix factorization model, for which the output for a pair $(u, i) \in U \times I$ is given by

$$\hat{y}(u,i) = \sum_{f=1}^{d} v_{u,f}\, v_{i,f},$$

where $v_{u,f}, v_{i,f}$ are the parameters of the model. We use the parameterization of the loss as proposed in [3]:

$$\operatorname*{argmin}_{V} \sum_{(u,i)\in H} (\hat{y}(u,i) - 1)^2 + \alpha \sum_{u\in U} \sum_{i\in I} \hat{y}(u,i)^2 + \lambda ||V||_F^2,$$

optimized with implicit alternating least squares [11]. The hyperparameters are $d = 16, \lambda = 10, \alpha = 0.2$.

Recommenders Y and Z are item based collaborative filtering algorithms [14]. There are many different variations of this algorithm, in particular how to generate the similarity matrix. We use the following definition. The basic similarity is cosine and we follow the suggestion by [1] to apply an exponent q to sharpen the similarity:

$$s_{i,j} = \left(\frac{|\{u : (u,i) \in H\} \cap \{u : (u,j) \in H\}|}{\sqrt{|\{u : (u,i) \in H\}|}\sqrt{|\{u : (u,j) \in H\}|}} \right)^q \tag{27}$$

Then we add the option to sparsify based on k-nearest neighbors:

$$s'_{i,j} = s_{i,j}\, \delta(i \in N_k(j))\, \delta(j \in N_{k'}(i)), \tag{28}$$

where $N_k(i)$ are the k closest items to i based on s. We allow both selecting for row or column neighbors. Then we apply row normalization:

$$s''_{i,j} = \frac{s'_{i,j}}{||\mathbf{s}'_i||}. \tag{29}$$

Finally the prediction is:

$$\hat{y}(u,i) = \frac{\sum_{j:(u,j)\in H} s''_{i,j}}{\sum_{j\in I} s''_{i,j}}. \tag{30}$$

Recommender Y uses the following hyperparameters: $q = 3, k = \infty, k' = \infty$. Recommender Z uses the following hyperparameters: $q = 1, k = \infty, k' = 10$.