Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)
Sister Conferences Best Papers Track

On Sampled Metrics for Item Recommendation (Extended Abstract)

Walid Krichene and **Steffen Rendle**
Google Research
{walidk,srendle}@google.com

Abstract

Recommender systems personalize content by recommending items to users. Item recommendation algorithms are evaluated by metrics that compare the positions of truly relevant items among the recommended 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 such sampled metrics and shows that they are inconsistent with their exact counterpart, in the sense that they do not persist relative statements, e.g., *recommender A is better than B*, not even in expectation. We show that it is possible to improve the quality of the sampled metrics by applying a correction. We conclude with an empirical evaluation of the naive sampled metrics and their corrected variants. 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 estimates.

1 Introduction

Item recommendation is, at its core, a retrieval task, where given a context, a catalogue of items is ranked and the top scoring ones are shown to the user. Ranking all items can be costly for large catalogues. Recently, it has become common in research papers to speed up evaluation by only ranking a small subset, consisting of the relevant items together with a random sample of irrelevant ones [He *et al.*, 2017; Ebesu *et al.*, 2018; Hu *et al.*, 2018; Yang *et al.*, 2018b; Yang *et al.*, 2018a; Krichene *et al.*, 2019; Wang *et al.*, 2019]. While sampled training is well-studied [Yu *et al.*, 2017], to the best of our knowledge, the implications of sampled *evaluation* have not been explored, and this work attempts to shed light on the topic. In particular, we show 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 a sampled metric, this does not imply that A has a better metric than B when the metric is computed exactly. This problem occurs even in expectation; i.e., with unlimited repetitions of the measurement. Moreover, the smaller the sample size, the less difference there is between different metrics, and in the small sample limit, all metrics collapse to the area under the ROC curve (AUC). This is particularly problematic because many ranking metrics are designed to focus on the top positions, which is not the case for AUC.

Our 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, readers should be aware that the reported metric has different characteristics than its name implies.

2 Evaluating Item Recommendation

We briefly recap the evaluation scheme for item recommendation that we investigate in this work. 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 position, $r(A, \mathbf{x}) \in \{1, \dots, n\}$, of the withheld relevant item within this ranking is computed – r will also be referred to as the *predicted rank*[2]. For example, $r(A, \mathbf{x}) = 3$ means for an instance $\mathbf{x}$ recommender A ranked the relevant item at position 3. Then, a metric M translates the predicted rank into a quality value. This process is repeated for a set of instances, D, and an average metric is reported: $\frac{1}{|D|} \sum_{\mathbf{x} \in D} M(r(A, \mathbf{x}))$. For convenience, we will omit the arguments $A, \mathbf{x}$ from $r(A, \mathbf{x})$ whenever the particular recommender, A, or instance, $\mathbf{x}$, is clear from context. We now recap some popular examples for metrics M:

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

$$\text{Prec}(r)_k = \delta(r \leq k)\frac{1}{k}, \tag{2}$$

$$\text{Recall}(r)_k = \delta(r \leq k), \tag{3}$$

$$\text{AP}(r)_k = \delta(r \leq k)\frac{1}{r}, \tag{4}$$

$$\text{NDCG}(r)_k = \delta(r \leq k)\frac{1}{\log_2(r + 1)}. \tag{5}$$

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

[2] For simplicity, we focus on the case where there is only one relevant item. See [Krichene and Rendle, 2020] for a more general case.

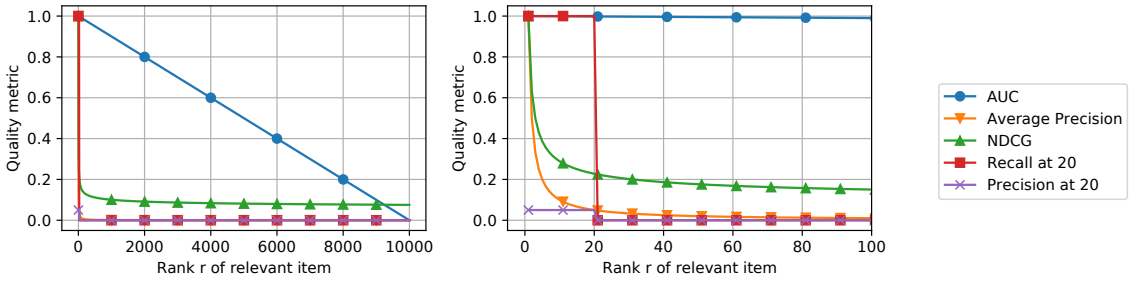

Figure 1: Visualization of metric vs. predicted rank for $n = 10,000$. The left side shows the metrics over the whole set of $10,000$ items. The right side zooms into the contributions of the top 100 ranks. All metrics besides AUC are top heavy and almost completely ignore the tail. This is usually a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

For metrics such as Average Precision and NDCG, it makes sense to also define their untruncated counterpart, i.e., for $k = n$: $\text{AP}(r) = \frac{1}{r}$ and $\text{NDCG}(r) = \frac{1}{\log_2(r+1)}$. 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., improving the ranking of a relevant item from position 101 to 100 is as valuable as an improvement from position 2 to 1. Table 1 shows a toy example for an evaluation.

3 Sampled Metrics

Ranking all items is expensive when the number of items, n, is large. Recently, it has become common to rank only a small set, consisting of the relevant item together with a random sample of m irrelevant ones. Let $\tilde{r}$ be the rank of the relevant item within this random set – note that $\tilde{r}$ is a random variable. The metric, M, is then computed on $\tilde{r}$ instead of r. Examples of work that use this sampling protocol include [He *et al.*, 2017; Ebesu *et al.*, 2018; Hu *et al.*, 2018; Yang *et al.*, 2018b; Yang *et al.*, 2018a; Krichene *et al.*, 2019; Wang *et al.*, 2019].

3.1 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. *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,*

$$\frac{1}{|D|}\sum_{\mathbf{x}\in D} M(r(A,\mathbf{x})) > \frac{1}{|D|}\sum_{\mathbf{x}\in D} M(r(B,\mathbf{x}))$$
$$\iff E\left[\frac{1}{|D|}\sum_{\mathbf{x}\in D} M(\tilde{r}(A,\mathbf{x}))\right] > E\left[\frac{1}{|D|}\sum_{\mathbf{x}\in D} M(\tilde{r}(B,\mathbf{x}))\right]. \quad (6)$$

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

We illustrate this on our toy example. Table 2 shows the sampled metrics for the example from Table 1. An evaluation with a sampled metric is a random process, so for a better understanding of its outcome, we repeat the evaluation 1000 times and report the average and standard deviation. Compared to the exact metrics in Table 1, the relative ordering of metrics 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. Only AUC is consistent for all m, and the expected metric is independent of sample size.

3.2 Rank Distribution Under Sampling

This section derives the distribution of the sampled rank $\tilde{r}$ and discusses expected metrics. 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:

$$p(j < r) = \frac{r-1}{n-1}. \quad (7)$$

For example, if r is at position 1, the likelihood of a random irrelevant being ranked higher is 0. If $r = n$, then the like-

	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 and $n = 10,000$ items. A *predicted rank* is the position where a relevant item was ranked by a recommender. Recommender C that has one highly ranked relevant item performs the best on the top heavy metrics AP, NDCG and Recall@10, while recommender A where relevant items score neither high nor low performs best on AUC.

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

Table 2: Metrics for the sampled evaluation protocol for the recommenders from Table 1. $m = 99$ 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. As can be seen, on sampled metrics the relative ordering of recommenders A, B, C is **not** preserved, except for AUC.

lihood 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 is 1, if all m samples are successful, the rank is $m + 1$. The expected value of the metrics under this distribution is

$$E[M(\tilde{r})] = \sum_{i=1}^{m+1} p(\tilde{r} = i) M(i). \tag{8}$$

Note that this is implicitly a function of r, m and n, 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. The observations are supported by a formal analysis in [Krichene and Rendle, 2020].

4 Corrected Metrics

This section investigates whether we can design a sampled metric $\hat{M}$, a function from $\{1, \dots, m + 1\}$ to $\mathbb{R}$, such that $\hat{M}(\tilde{r})$ provides a good estimate of $M(r)$.

4.1 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 3.2 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:

$$\hat{M}(\tilde{r}) = M\left(1 + \frac{(n-1)(\tilde{r}-1)}{m}\right). \tag{9}$$

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 $\lfloor \cdot \rfloor$.

4.2 Bias-Variance Trade-off

Another criterion one may seek to optimize is the average bias of $\hat{M}(\tilde{r})$, that is, $\sum_r (E[\hat{M}(\tilde{r})|r] - M(r))^2$. One potential issue with the minimal bias estimator is that it could have high variance, which we observe numerically in Section 5. In order to alleviate this problem, we can introduce a variance term. Since $\hat{M}$ is a function from $\{1, \dots, m + 1\}$ 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 bias-variance trade-off

$$\operatorname*{argmin}_{\hat{M} \in \mathbb{R}^{m+1}} \sum_{r=1}^{n} \left((E[\hat{M}_{\tilde{r}}|r] - M(r))^2 + \gamma \operatorname{Var}[\hat{M}_{\tilde{r}}|r] \right), \tag{10}$$

where γ is a positive constant that controls the trade-off. Eq. (10) is a regularized least squares problem with a closed form solution, see [Krichene and Rendle, 2020] for additional discussion.

5 Experiments

In [Krichene and Rendle, 2020], an experimental study is performed on instances of real recommender algorithms and a real dataset. We summarize some of the findings here and refer to the full paper for details. The study uses the sampled item recommendation evaluation protocol from [He *et al.*, 2017]. The data comes from the movie recommender Movielens [Harper and Konstan, 2015] where users rate movies.

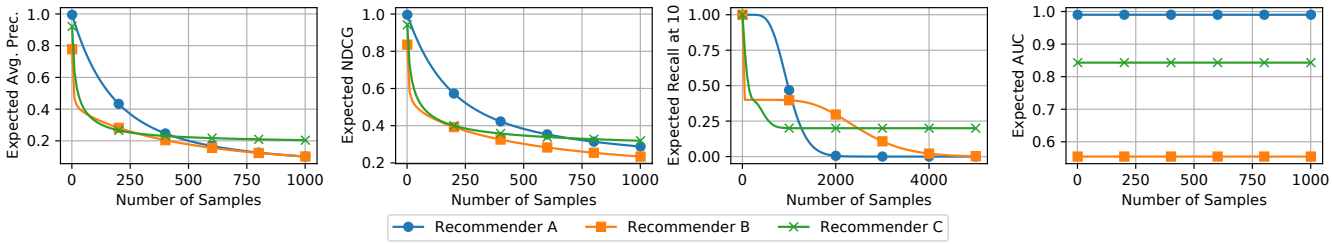

Figure 2: Expected sampled metrics for the running example (Tables 1 and 2) while increasing the sample size. For Average Precision, NDCG and Recall, even the relative order of recommender performance changes with the number of samples. That means, conclusions drawn from a subsample are not consistent with the true performance of the recommender.

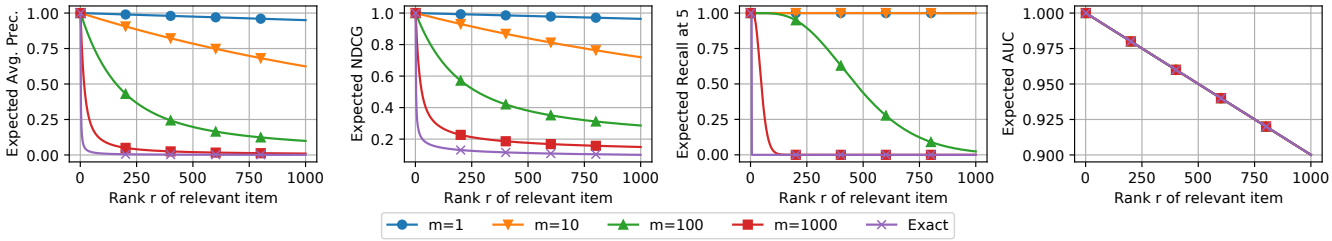

Figure 3: Characteristics (compare to Fig 1) of sampled metrics with a varying number of samples, m. Sampled Average Precision, NDCG and Recall change their characteristics substantially compared to exact computation of the metric. Even large sample sizes ($m = 1000$ samples of $n = 10000$ items) show large bias. Note this plot zooms into the top 1000 ranks out of $n = 10000$ items.

Rank distribution. We find that real algorithms may result in different rank distributions, for example in the experiment, one of the recommenders is best in the top 10 but has poorer performance at higher ranks, while another recommender is more balanced and puts only few items at poor ranks.

Sampled Metrics. The experiment confirms that sampled metrics can become inconsistent with their exact counterpart. We find that uncorrected sampled metrics have low standard deviation, so the issue is not that of variance, but is due to the bias in the sampled metrics. In particular, for the recommender instances in the experiments, if the study would compare the recommenders only on the sampled metrics, it would draw the wrong conclusion for top-heavy metrics such as Recall, NDCG and AP, even with unlimited repetitions of the experiment. The worst recommender would be found to be the best one.

Corrected Metrics. Finally, the study shows that correction strategies proposed in Section 4 improve the metric estimates. Even though they often have an increased variance, corrected metrics are better at identifying the true order of algorithms. We repeat measurements and report the proportion of times the correct order is identified by each method and for each pairwise comparison, and find that despite the higher variance, corrected metrics have a higher success rate. In particular, the simple *rank estimate* (eq. 9), while trivial to implement (i.e., upscaling the rank before applying the metric), already gives a notable improvement. Other corrections such as *bias-variance* (eq. 10) yield better results, but can be more difficult to implement and apply because the bias-variance trade-off γ needs to be configured carefully.

6 Concluding Remarks

This work seeks to bring attention to some issues with sampling of evaluation metrics for item recommendation. 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. This is mostly due to the large bias introduced by sampling.

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. In this case it is important to rerun the experiment with different samples (e.g., different random seeds). Common practices already have several sources of variance (e.g. due to dataset splits, random initialization), and corrected metrics will introduce another source of variance. This means that it may be harder to find "statistically significant" differences between two recommenders. While corrected metrics are preferable to uncorrected ones, it is important to keep in mind that they are 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.

Acknowledgements

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

References

[Ebesu *et al.*, 2018] Travis Ebesu, Bin Shen, and Yi Fang. Collaborative memory network for recommendation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 515–524. ACM, 2018.

[Harper and Konstan, 2015] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182, 2017.

[Hu *et al.*, 2018] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 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*, KDD '18, pages 1531–1540. ACM, 2018.

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

[Krichene *et al.*, 2019] Walid Krichene, Nicolas Mayoraz, Steffen Rendle, Li Zhang, Xinyang Yi, Lichan Hong, Ed Chi, and John Anderson. Efficient training on very large corpora via gramian estimation. In *International Conference on Learning Representations*, 2019.

[Wang *et al.*, 2019] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, AAAI '19, pages 5329–5336, 2019.

[Yang *et al.*, 2018a] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, and Deborah Estrin. Openrec: A modular framework for extensible and adaptable recommendation algorithms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 664–672. ACM, 2018.

[Yang *et al.*, 2018b] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 279–287. ACM, 2018.

[Yu *et al.*, 2017] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. Selection of negative samples for one-class matrix factorization. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 363–371, 2017.