KDD2020最佳论文: 关于个性化排序任务评价指标的大讨论

张小磊
北京交通大学 计算机科学与技术博士在读

前段时间还跟实验室同学专门讨论了下关于个性化排序中的评价指标问题，即我们在实验的过程中究竟使用哪一种实验设置才能较好的反映模型的泛化能力，比如采用全量测试数据进行排序，但该方法需要的测试时间较长；比如使用采样的测试样本进行排序评估，该方法虽然评估时间大大缩短但不能完全反映模型性能。如今Rendle大佬通过实验的方式告诉了我们答案，并且凭借该主题获得了KDD2020的最佳论文，祝贺！

众所周知，推荐系统主要有两大任务：评分预测和个性化排序。其中，针对评分预测任务的评判指标主要是均方误差（MSE）、均方根误差（RMSE）和平均绝对误差（MAE）等回归指标。这些指标的评价相对来说复杂度较低，因此对于全量测试数据进行评估相对可行。对于个性化排序任务需要在给定上下文的情况下对大量的项目候选进行排序，因此需要利用平均准确率（MAP）、归一化折损累计增益（NDCG）等排序指标来评估模型的性能。如果大家想了解更多排序模型知识可以移步推荐系统中排序学习的三种设计思路。

目前，主流的个性化排序任务（Item Recommendation）的文献为了加速评价指标的计算，经常利用采样的指标（Sampled Metrics）进行评价，即针对待测试的正样本和随机出来的较小规模的负样本进行排序，比如在测试阶段对一个正样本和从大量候选集采样出来的99个负样本进行排序，然后计算该样本相对于负样本的排序位置进行性能评估。虽然这种实验设置可以一定程度

而且**对于非常小的抽样规模，所有指标都会塌陷为AUC指标。**因此，论文提出了一种改进的采样评价指标用来提高评价质量。最后，该论文建议评价的时候尽量不要采样，如果不听话非要采样那就用所提出的修正的采样指标来提高评价质量。

来，让我们先来熟悉下常用的排序指标，即AUC，Precision，Recall，AP和NDCG。其中，

n 为全部物品个数， R 为预测的列表结果， $|R|$ 为预测的样本个数， r 代表该物品所在的位置， k 为设置的预测截断个数， m 为测试时采样的负样本个数。

AUC衡量了相关项目排在非相关项目前边的可能性。

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

Precision衡量了在前k个预测物品中相关物品的比例。

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

Recall代表预测召回的物品中排在前k位置物品的比例。

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

Average Precision表示对于前边Precision指标的平均。

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

NDCG为归一化的折损累计收益，通过在分母引入位置收益来表示排在前边并且收益大的项目获得的收益较高。

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

本，以此来进行排序，看最终该正样本排在了什么位置。因此上文的精确采样可以表示为下图所示的简化形式。

$$\mathrm{AUC}(r)_n = \frac{n-r}{n-1}$$
$$\mathrm{Prec}(r)_k = \delta(r \le k)\frac{1}{k}$$
$$\mathrm{Recall}(r)_k = \delta(r \le k)$$
$$\mathrm{AP}(r)_k = \delta(r \le k)\frac{1}{r}$$
$$\mathrm{NDCG}(r)_k = \delta(r \le k)\frac{1}{\log_2(r+1)}$$

接下来主要介绍下论文中的实验结果分析与结论。

下图1展示的是将正样本随着排序位置的变化所产生的评价指标的变化。左图是针对所有的候选集来说的，右图是针对Top100来说的。从左图可以看出AUC是与排序位置无关的指标，随着排名逐渐靠后，排序指标线性的递减。也就是说把正样本从排名100移到101位的变化跟把排名从第2位移到第1位一样；平均准确率AP的分数衰减的最明显，例如在排名第1位的价值是排名第2的两倍；右图展示了各种指标在Top100的指标变化，**可见除了AUC以外，其他指标都对排序位置比较敏感。**

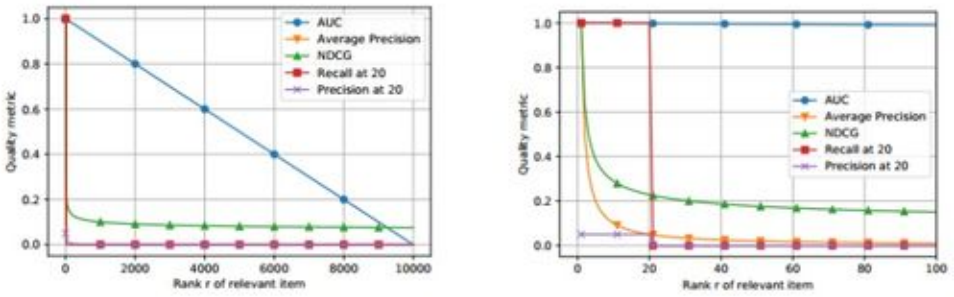

Figure 1: Visualization of metric vs. predicted rank for $n = 10,000$. The left side shows the metrics over the whole set of 10,000 items. The right side zooms onto the contributions of the top 100 ranks. All metrics besides AUC are top heavy and almost completely ignore the tail. This is usually a desirable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

对于排序评价指标的选择取决于是否位置敏感，即由于用户的注意力有限以及网站或者App有限的展示资源，人们大多比较关心排在头部的物品之间的相对位置，而忽略尾部的项目排序位置，因此对于测试阶段我们需要强调头部效应。而在训练阶段我们需要尽量打消这样的基于位置的偏见（Position bias），尽可能的还原用户点击该物品是真的处于喜欢，而非仅仅因为排在了头部显眼的位置。我喜欢你，不仅仅是因为你出现在了我眼前，而更是因为你的内在。

下表展示了3个推荐算法A，B，C预测结果不同而产生的关于AUC，AP，NDCG和Recall的评果。表1是针对5个实例的精确评价，表2是采样过后进行的评价。可见只有AUC这种位置不敏

	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.

另外，论文还针对采样个数关于评价指标的变化进行了实验。实验结果出现群魔乱舞的现象。发现随着评价阶段负采样个数的增加，原来性能优越的算法A出现性能恶化，最终被算法C打败的情况。**可见，只针对一个负采样个数来作为最终模型的性能评价有失公平。**这么一想，咱的破模型没准在某个负采样个数的设置下可能打败著名的N某F。

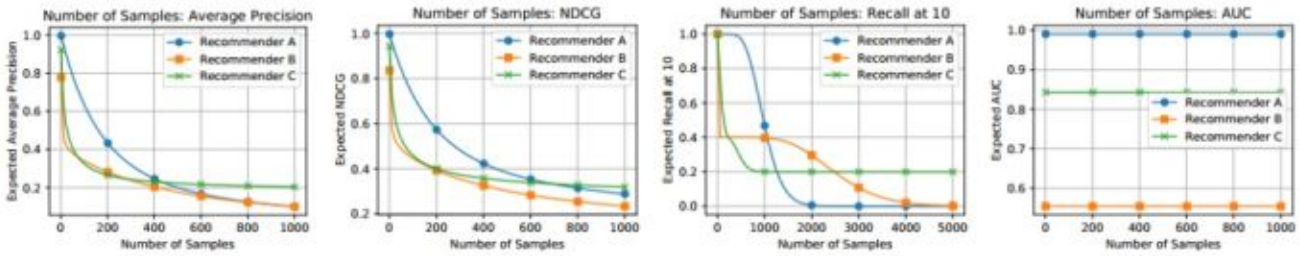

Figure 2: Expected sampling metrics for the running example (Section 3.2 and 4.2) while increasing the number of samples. For Average Precision, NDCG and Recall, even the relative order of recommender performance changes with the number of samples. That means, conclusions drawn from a subsample are not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

另外，论文对于采样指标中不同的采样个数对结果的影响与精确的指标做了相关对比实验。可见不同的采样个数与精确的指标之间差距较大，并且即使采样个数足够大，仍然与精确的评价指标之间存在较大偏差。所以，**只利用某一种负采样个数进行性能评估是具有偶然性的，但往往必然的结果是与真正的评价效果相差很远。**

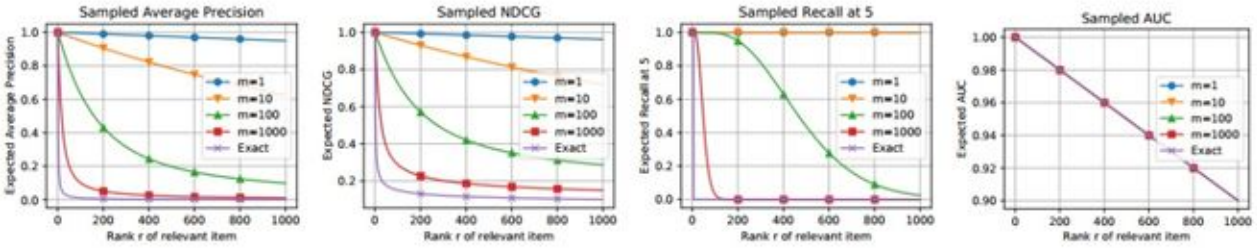

Figure 3: Characteristics of a sampled metric with a varying number of samples. Sampled Average Precision, NDCG and Recall change their characteristics substantially compared to exact computation of the metric. Even large sample sizes (e.g. 1000 samples of $n = 10000$ items) show large bias. Note this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

知乎 首发于 推荐系统传送门

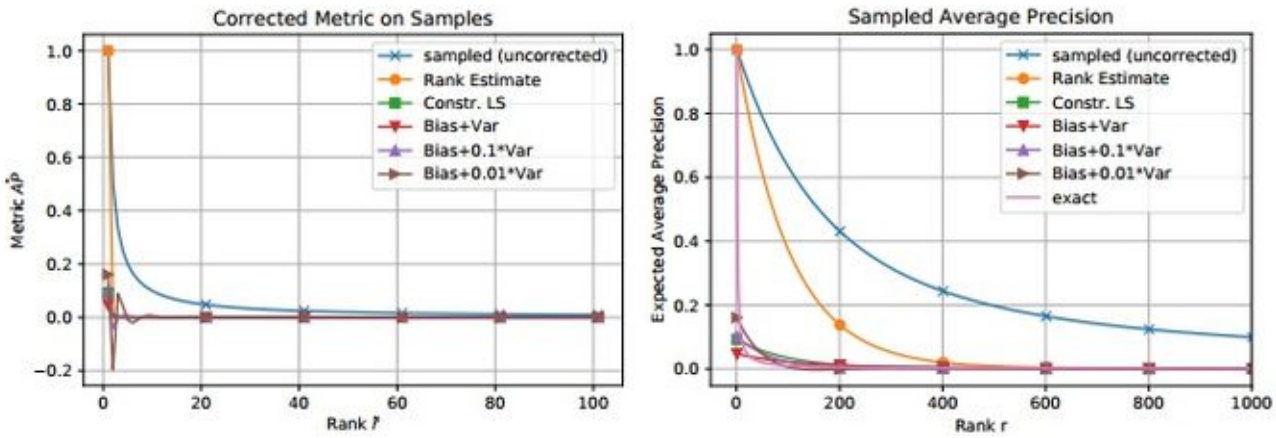

Figure 4: Evaluating the corrected metric $\hat{AP}$ on a sample of $m = 100$ items (left) is equivalent to measuring the metric on the full item set of $m = 10,000$ (right). Different choices of correction algorithms are plotted.

最后，大佬给出了一些做实验的建议。

抽样指标可能无法很好地指示该指标下推荐算法的真实性能。对于未校正的指标，这主要是由于采样引入的较大偏差造成的。使用校正方法，可以减少这种偏差，但要付出更高的方差代价。如果论文中确实需要使用抽样指标，并且仍对指标的真实性能感兴趣，建议使用本文提出的校正方法。在这种情况下，请务必使用不同的样本（例如，不同的随机种子）重新进行实验。尽管这种改进的评价指标优于未校正的采样指标，但由于偏差，它仍然倾向于得出错误的结论。所以只有完全避免抽样，才能消除这种偏差。

更多关于论文细节，请阅读原文。

发布于 2020-08-30

推荐系统 机器学习 深度学习（Deep Learning）

文章被以下专栏收录