最全面的推荐系统评估方法介绍

原创 子墨 搜索与推荐Wiki 2020-11-11

收录于话题
#推荐相关笔记

32个

▼ 往期精彩回顾 ▼

多角度审视推荐系统

Embedding技术在推荐系统中的应用

特征工程 | 文本特征处理的四大类主流方法

编辑：子墨

来源：《深度学习推荐系统》笔记，并进行补充和说明

推荐系统覆盖于生活中的各个方面，无论是电商购物，还是内容咨询，都离不开它的身影，作为一名推荐算法从业者，深知做好推荐系统的必要性，那么做好推荐系统的评估就显得至关重要了，其主要体现在：

- 推荐系统评估所采用的指标直接决定了推荐系统的优化方向是否客观合理
- 推荐系统评估是机器学习团队与其他团队沟通合作的接口性工作
- 推荐系统评估指标的选取直接选定了推荐系统是否符合公司的商业目标和发展愿景

做好推荐系统的评估的前提是必须要搞明白评估指标有哪些？分别适用于什么场景？怎么选择？所以才有了下面的内容

无论你是否对推荐系统的评估有个系统的了解，都建议再读读，权当作是回味复习或者是增强记忆

那么各位看官开始阅读吧！

离线评估的主要方法

Holdout检验

Holdout检验是基础的离线评估方法，它将原始的样本集合随机划分为训练集和验证集两部分，比如70%训练集，30%测试集（但现在很多机器学习框架、深度学习框架中都增加了验证集，即将整个数据集分成三份，70%训练集，10%验证集，20%测试集）。

Holdout检验的缺点也很明显，即在验证集上计算出来的评估指标与训练集和测试机的划分有直接关系，如果仅进行少量Holdout检验，则得到的结论存在很大的随机性（在划分数据集的时候尽量保证其随机性）。

交叉检验

1、k-fold交叉验证

先将全部样本划分成 k 个大小相等的样本子集，依次遍历这k 个子集，每次都把当前子集作为验证集，其余所有子集作为训练集，进行模型的训练和评估，最后将所有k次的评估指标的平均值作为最终的评估指标，在实际经验中，k 经常取值为 10。

2、留一验证

每次留下1个样本作为验证集，其余所有样本作为测试集，样本总数为n，依次遍历所有n 个样本，进行n次验证，再将评估指标求平均得到最终指标。在样本总数较多的情况下，留一验证法的时间开销极大，事实上，留一验证是留p 验证的特例，留p 验证是指每次留下p 个样本作为验证集，而从 n 个元素中选择 p 个元素 有 C_n^p 种可能，因此它的时间开销远远高于留一验证，故很少在实际中使用。

3、自助法

不管是holdout检验还是交叉检验，都是基于划分训练集和测试集的方法进行模型评估的，当样本规模比较小时，将样本集进行划分，会进一步缩小训练集，有影响模型的训练效果。

自助法（Bootstrap）是基于自助采样法的检验方法：对于总数为n的样本集合，进行n 次有放回的随机抽样，得到大小为n的训练集，在n次采样过程中，有的样本会被重复采样，有的样

本没有被抽出过，将这些没有被抽出的样本作为验证集进行模型验证，就是自助法的验证过程。

离线评估的主要指标

准确率

分类准确率是指分类正确的样本占总样本个数的比例：

$$accuracy = \frac{n_{correct}}{n_{total}}$$

$n_{correct}$ 为被正确分类的样本个数， n_{total} 为总样本个数，准确率是分类任务中比较直观的评价指标，但其优缺点也明显

- 优点：解释性强
- 缺点：类别分布不均匀时，占比大的类别往往成为影响准确率的主要因素（极端的情况比如正样本 1%，负样本 99%时）

精确率和召回率

- 精确率（Precision）是分类正确的 正样本 个数占 分类器判定为正样本 的样本个数的比例
- 召回率（Recall）是分类正确的 正样本个数 占 真正正样本 个数的比例

排序模型中，通常没有一个确定的阈值把预测结果直接判定为正样本还是负样本，而是采用 Top N 排序结果的精确率（Precision@N）和召回率（Recall@N）来衡量排序模型的性能，即认为模型排序的Top N的结果就是模型排定的正样本，然后计算精确率和召回率。

精确率和召回率是矛盾统一的两个指标：为了提高精确率，分类器需要尽量再“更有把握时”才把样本预测为正样本，但往往因为过于保守而漏掉很多“没有把握”的正样本，导致召回率降低。

因此使用F1-score进行调和（也叫F-measure），定义为：

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

均方根误差

Root Mean Square Error，RMSE 经常被用来衡量回归模型的好坏，使用点击率预估模型构建推荐系统时，推荐系统预测的其实是样本为正样本的概率，RMSE被定义为：

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - y')^2}{n}}$$

y_i 为第i 个样本的真实值，y'为第i 个样本的预测值，n 为样本的个数。

RMSE的缺点：一般情况下能够很好的反映回归模型预测值与真实值的偏离程度，但在实际应用时，如果存在个别偏离程度非常大的离群点，那么即使离群点的数量非常少，也会让RMSE指标变得很差

为了解决这个问题，引入了鲁棒性更强的平均绝对百分比误差（Mean Absolute Percent Error，MAPE）进行类似的评估，MAPE定义如下：

$$MAPE = \sum_{i=1}^{n} |\frac{y_i - y'}{y_i}| * \frac{100}{n}$$

相比RMSE，MAPE相当于把每个点的误差进行了归一化，降低了个别离群点带来的绝对误差的影响。

对数损失函数

LogLoss，在一个二分类问题中，LogLoss定义为：

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}(y_i log P_i + (1 - y_i) log(1 - P_i))$$

其中y_i 为输入实例x_i 的真实类别，p_i 为预测输入实例x_i是正样本的概率，N 为样本总数。

> LogLoss 是逻辑回归的损失函数，大量深度学习模型的输出层是逻辑回归或softmax，因此采用LogLoss作为评估指标能够非常直观的反映模型损失函数的变化，站在模型的角度

> 来讲，LogLoss非常适于观察模型的收敛情况。

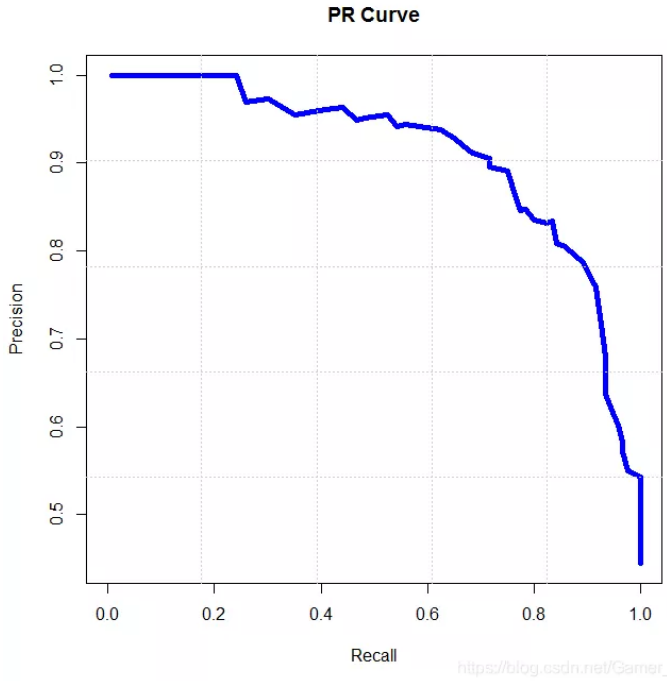
关于对数损失函数和交叉熵损失函数的介绍可以参考：从极大似然到对数损失函数和交叉熵损失函数，以及对数损失优化取值范围

直接评估推荐序列的离线指标

P-R曲线

P-R曲线的横轴是召回率，纵轴是精确率，对于一个排序模型来说，其P-R曲线上的一个点代表在某一阈值下，模型将大于该阈值的结果判定为正样本，将小于该阈值的结果判定为负样本时，排序结果对应的召回率和精确率

整体P-R曲线是通过从高到低移动正样本阈值生成的，如下所示：



P-R曲线下的面积（Area Under Curve， AUC）能够量化P-R曲线的优劣，AUC越大，排序模型的性能越好。

ROC曲线

ROC曲线的全称时 the Receiver Operating Characteristic曲线，中文译为「受试者工作特征曲线」，ROC曲线最早诞生于军事领域，而后在医学领域应用甚广，「受试者工作特征曲线」也来源于该领域。

ROC曲线的横坐标时 False Posotive Rate（FPR， 假阳性率），纵坐标时True Positive Rate（TPR，真阳性率），FPR和TPR的计算方法如下：

$$FPR = \frac{FP}{N}, TPR = \frac{TP}{P}$$

上式中 P 时真实的正样本数量，N 是真实的负样本数量，TP 指的是P个正样本中被分类器预测为正样本的个数，FP指的是

这里涉及到混淆矩阵，如下所示（关于AUC的具体计算方式可以参考：模型评估中的AUC是怎么计算的 ，描述的比较详细）：

		预测值	
		0	1
真实值	0	40	60
	1	60	40

		预测值	
		0	1
真实值	0	TN（True negative）	FP（False positive）
	1	FN（False negative）	TP（True positive）

ROC曲线的绘制和P-R曲线一样，通过不断移动模型正样本阈值生成的，ROC曲线下的面积就是AUC（绘制的过程参考内容：模型评估中的AUC是怎么计算的？）

平均精度均值

平均精度均值（Mean Average Precision，mAP）是另一个在推荐系统、信息检索领域常用的评估指标，该指标其实是对平均精度（Average Precision，AP）的再次平均。

假设推荐系统对某一用户测试集的排序结果如下所示：

推荐序列	**N=1**	**N=2**	**N=3**	**N=4**	**N=5**	**N=6**
真实标签	1	0	0	1	1	1

其中，1代表正样本，0代表负样本

那么对于上述的序列，precision@N分别是多少呢?

推荐序列	**N=1**	**N=2**	**N=3**	**N=4**	**N=5**	**N=6**
真实标签	1	0	0	1	1	1
precision@N	1/1	1/2	1/3	2/4	3/5	4/6

AP的计算只取正样本处的precision进行平均，即 AP= (1/1 + 2/4 + 3/5 + 4/6) =0.6917

那么mAP是什么呢?

如果推荐系统对测试集中的每个用户都进行样本排序，那么每个用户都会计算出一个AP值，再对所有用户的AP值进行平均，就得到了mAP，也就是mAP是对精确度平均的平均。

需要注意的是，mAP的计算和P-R曲线、ROC曲线的计算方法完全不同，因为mAP需要对每个每个用户的样本进行分用户排序，而P-R曲线和ROC曲线均是对全量测试样本进行排序。

合理的选择评估指标

除了上述介绍的几种评估指标，推荐系统的评估指标还包括：

- 归一化折损累计收益（Normalized Discounted Cumulative Gain，NDCG）
- 覆盖率（Coverage）
- 多样性（diversity）

虽然离线评估推荐系统有很多指标，但我们进行离线模型的评估时并不能使用全部的指标，没必要陷入「完美主义」和「实验室思维」的误区，选择2-4个适合自己业务的指标进行优化即可。

NDCG

上文中提到了折扣累计收益，这个是搜索排序中使用比较广的评估指标，因此这里也补充介绍一下。NDCG，归一化折损累计收益，这里边涉及三个概念，分别是CG、DCG、NDCG，依次来理解。

> From：https://www.cnblogs.com/by-dream/p/9403984.html

CG

CG，cumulative，是DCG的前身，只考虑到了相关性的关联程度，没有考虑到位置的因素，它是一个搜索结果相关性分数的综合，指定位置p上的CG为：

$$CG_p = \sum_{i=1}^{p} rel_i$$

rel_i 代表 i 这个位置上的相关度。

比如搜索"推荐系统"图书时，最理想的结果时R1、R2、R3，但出现的结果是 R2、R3、R1，CG值是没有变化的。

DCG

DCG， Discounted的CG，就是在每一个CG的结果上除以一个折损值，目的是为了让排名越考前的结果越能影响最后的结果，假设排序越靠后，价值越低，那么到第i个位置时，价值为 $\frac{1}{log_2(i+1)}$，那么第i个结果产生的效益就是 $rel_i * 1/log_2(i+1)$，所以DCG表达式为：

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} = rel_i + \sum_{i=2}^{n} \frac{rel_i}{log_2(i+1)}$$

另外一种比较常用的公式，增加相关度影响比重的DCG计算方式是：

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rei_i} - 1}{log_2(i+1)}$$

百科中写到后一种更多用于工业。当然相关性值为二进制时，即 rel_i在$0, 1$，二者结果是一样的。当然CG相关性不止是两个，可以是实数的形式。

NDCG

NDCG，归一化的DCG，由于搜索结果随着检索词的不同，返回的数量是不一样的，而DCG是一个累加的值，没法针对两个不同的检索结果进行归一化出力，这里是除以IDCG。

$$NDCG_p = \frac{DCG}{IDCG_p}$$

IDCG为理想情况下的最大的DCG的值，为：

$$IDCG_p = \sum_{i=1}^{|REL|} \frac{2^{rei_i} - 1}{log_2(i+1)}$$

其中$|REL|$ 表示，结果按照相关性从大到小的顺序排序，取前p个结果组成的集合，也就是按照最优的方式对结果进行排序。

更接近线上环境的离线评估方法-Replay

模型评估的逻辑闭环

- 1、什么样的模型是好模型？「对业界来说，能更好的实现公司商业目标的模型是好模型」
- 2、如何评估模型的商业价值和商业指标？「线上真实环境A/B测试，能够直接评估模型的商业指标」
- 3、如何在离线环境下得到线上A/B测试的评估指标？「能够让离线评估结果更接近线上评估结果的离线评估方法是好的评估方法」
- 4、什么是好的离线模型评估方法？「能够正确评估模型好坏的办法是好的离线评估方法」
- 5、什么样的模型是好模型？「 又回到1了，这就是评估模型的逻辑闭环 」

动态离线评估方法

传统离线评估方法和动态离线方法对比

- 传统离线方法：模型不会随着评估的进行而更新，假设用一个月得测试数据评估一个推荐系统，如果评估过程是静态的，这就意味着当模型对月末得数据进行预测时，模型已经停止更新近30天了，这不仅不符合工程实践，而且会导致模型效果评估得失真
- 动态离线评估方法：先根据样本产生时间对测试样本由早到晚进行排序，再用模型根据样本时间依次进行预测，在模型更新的时间点上，模型需要增量学习更新时间点前的测试样本，更新后继续进行后续的评估。

毫无疑问，动态评估的过程更接近真实的线上环境，评测结果也更接近客观情况，如果模型更新的频率持续增加，快到接收到样本就更新，整个动态评估的过程也变成逐一样本回放的精准线上仿真过程，这就是经典的仿真式离线评估方法-Replay。

Replay方法不仅适用于几乎所有推荐模型的离线评估，而且是强化学习类模型唯一的离线评估方法。

Replay的实际实现中有一点需要特别注意的是：
样本中不能包含任何「未来信息」，要避免数据穿越的现象发生

A/B测试与线上评估

上文介绍的离线评估指标无法还原真实的线上环境，几乎所有的互联网公司，线上A/B 测试都是验证新模块、新功能、新产品是否有效的主要方法。

什么是A/B测试

又称「分流测试」或「分桶测试」，是一个随机实验，通常被分为实验组和对照组。利用控制变量法，保持单一变量进行A、B两组的数据对比，并得到结论。

线上A/B测试无法被替代的原因主要有以下三点：

- 离线评估无法完全消除数据有偏（data bias）现象的影响，因此得到的离线评估结果无法完全替代线上评估结果
- 离线评估无法完全还原线上的工程环境，比如请求延迟、数据丢失、标签数据缺失等，离线评估比较理想化，结果存在失真现象
- 线上系统的某些商业指标再离线评估中无法计算

A/B测试的分桶原则

需要注意样本等独立性和无偏性，同一用户在测试的全程中只能被分到同一个桶中。

在实际的场景中，同一App或者网站需要进行多组不同类型的A/B测试，统同一业务的不同模块也会进行A/B测试（比如推荐系统中的召回层、排序层、展示层等），这种情况下不同层之间势必会产生干扰，同层之间也可能因为分流策略不当导致指标失真。

谷歌在其实验平台论文：Overlapping Experimen Infrastructure: More, Bette, Faster Experimentation 详细介绍了实验流量分层和分流的机制。A/B测试分流和分层的机制可以概括为：

- 层与层之间的流量 正交 ，即层与层之间的独立实验的流量是正交的，即实验中每组的流量穿越该层后，都会被再次随机打散，且均匀的分布再下层的每个实验中
- 同层之间的流量 互斥 ，即
  - 同层之间进行多组A/B测试，不同测试之间的流量是不重叠的
  - 一组A/B测试中实验组和对照组的流量是不重叠的，是互斥的

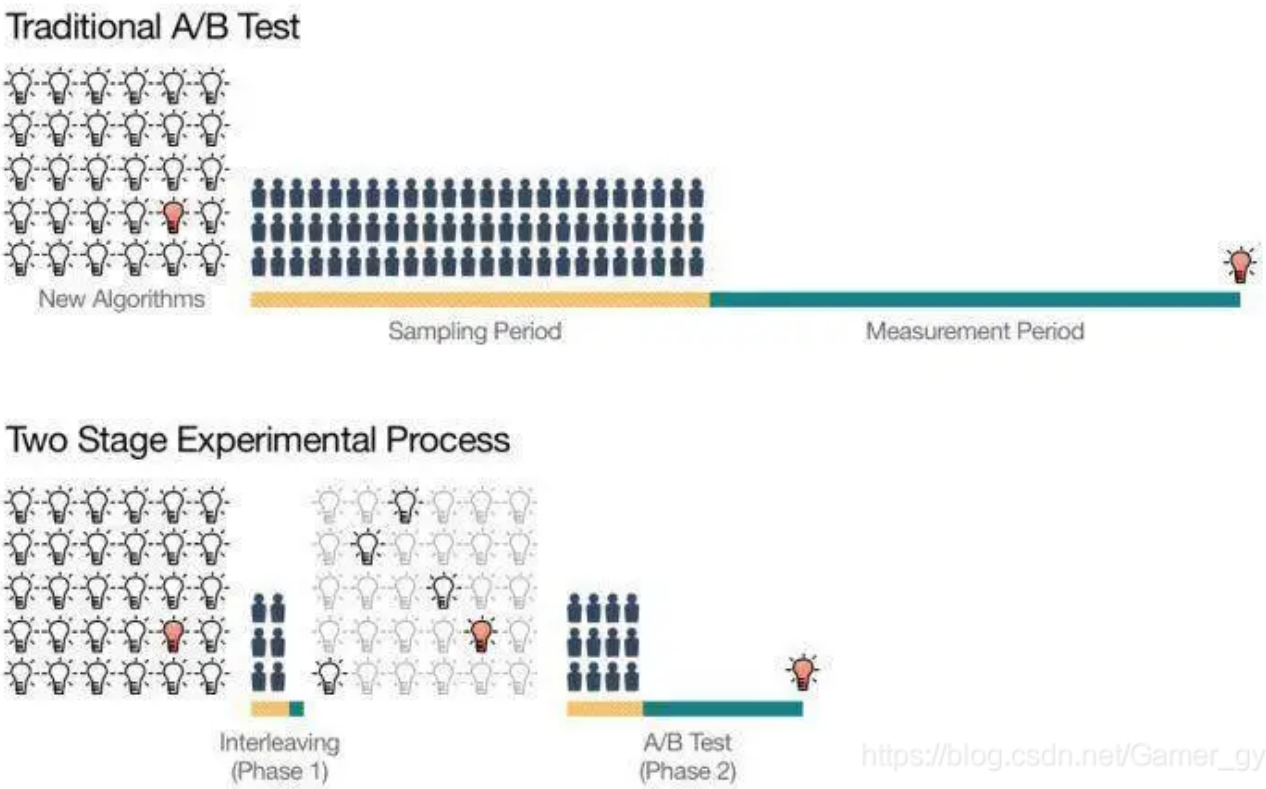
线上A/B测试的评估指标

不同业务背景关注的指标可能不一样，同一类型不同模块的业务下关注的指标也不一样，电商中经常关注的是：点击率、转化率、下单率、GMV、复购率等，娱乐咨询类平台关注的是：点击率、阅读时长、留存率等。

在进行A/B测试时，进行指标的对比和模型策略等的验证是比较有说服力的。

Interleaving线上评估方法

2013年微软提出了Interleaving线上评估方法，被当作时线上A/B测试的预选阶段进行候选算法的快速筛选，从大量初始想法中筛选出少量优秀的推荐算法，再对缩小的算法集合进行传统的A/B测试，以侧拉他们对用户行为的长期影响。



使用 Inter leaving 进行快速线上测试。用灯泡代表候选算法。其中，最优的获胜算法用红色表示。Interleaving 能够快速地将最初的候选算法集合进行缩减，相比传统的 AB Test 更快地确定最优算法。

A/B测试存在的统计学问题

A/B测试虽然是样本进行随机分配，但是难免会存在分布不均匀得情况，我们都知道二八原则，当对平台用户进行分流时，没有办法保证活跃用户也能被均分，因此一种可行的方法就是不对测试人群进行分组，而是让所有测试者都可以自由的选择要测试的物品，在实验结束时，统计每个人选择不同物品的比例，进行相关的数据统计（有点像做选择题哈哈），这种方案的优点在于：

- 消除了A/B测试者自身属性分布不均的问题

- 通过给予每个人相同的权重，降低了活跃用户对结果的影响

这种不区分A/B组，而是把不同的被测对象同时提供给受试者，最后根据守试者的选择得出评估结果的方法称为——Interleaving方法。

有关 Interleaving评估方法的更多细节内容参考：
https://netflixtechblog.com/interleaving-in-online-experiments-at-netflix-a04ee392ec55

Interleaving 方法的优缺点

优点：

- 所需样本少
- 测试速度快
- 结果与A/B测试无明显差异

缺点：

- 工程实现的框架较A/B测试复杂，实验逻辑和业务逻辑纠缠在一起，业务逻辑会被干扰
- Interleaving方法只是对"用户对算法推荐结果偏好程度"的相对测量，不能得出一个算法真实的表现，如果需要知道某个算法的具体指标提升，不适合使用Interleaving

Interleaving+A/B测试两阶的实验结构是比较合适和完善的，但随之而来的成本也会增加，企业在具体进行使用时，要结合具体的情况进行衡量，毕竟适合自己的才是最好的，另外对于推荐系统中的召回阶段，同批次并没有产出那么多的召回源，往往是逐个增加与迭代的过程，因此使用Interleaving方法也许并不合适，但是如果把Interleaving的思想和Bandit结合的话，说不定可以摩擦出更多火花

Over！内容到此结束，更多精彩内容，关注「搜索与推荐Wiki」不容错过，如果你觉得文章内容对你有帮助，点个 赞 、 在看 、 分享 再走吧！