搜索推荐评估指标—MRR、MAP和NDCG

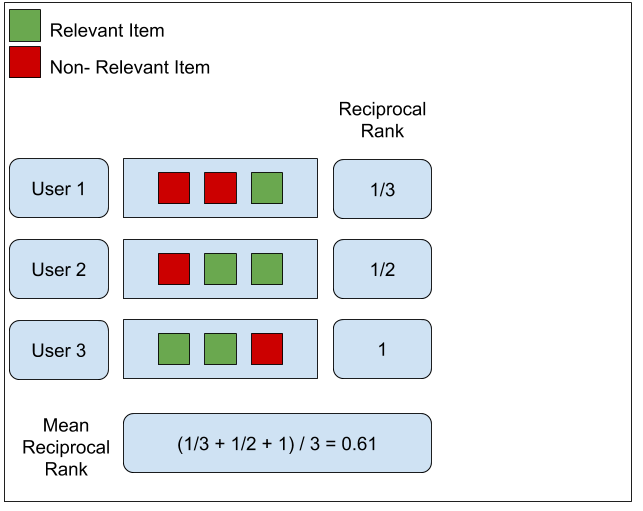
搜索和推荐系统需要把相关的物品放在推荐列表的最前面，用户不会通过滚动浏览200个条目来找到他们最喜欢物品。本文总结搜索和推荐系统中常用的三种评估指标：MRR、MAP和NDCG。

# 1 平均排名的倒数（MRR）

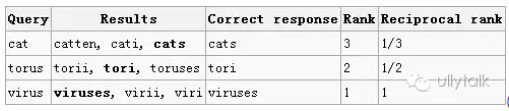
## 1.1 原理

MRR（Mean Reciprocal Rank）是最简单的评估指标，它的中文名称是平均排名的倒数。它试图度量“第一个相关的物品在哪里？”，把标准答案在被评价系统给出结果中的排序取倒数作为它的准确度，再对所有的问题取平均。

假设有以下三个针对三个用户的推荐列表。可以通过查找每个列表中第一个相关物品的排名来计算每个用户的倒数。然后我们对所有用户做一个简单的平均。



再比如：有3个query如下图所示：



(黑体为返回结果中最匹配的一项)

可计算这个系统的MRR值为：(1/3 + 1/2 + 1)/3 = 11/18=0.61。

MRR越接近1模型效果越好。

## 1.2 MRR的优点

* 该方法计算简单，解释简单。
* 这种方法高度关注列表的第一个相关元素。它最适合有针对性的搜索，比如用户询问“对我来说最好的东西”。
* 适用于已知项目搜索，如导航查询或寻找事实。

## 1.3 MRR的缺点

* MRR指标不评估推荐项目列表的其余部分。它只关注列表中的第一个项目。
* 它给出一个只有一个相关物品的列表。如果这是评估的目标，那找个度量指标是可以的。
* 对于想要浏览相关物品列表的用户来说，这可能不是一个好的评估指标。用户的目标可能是比较多个相关物品。

# 2 平均精度均值（MAP）

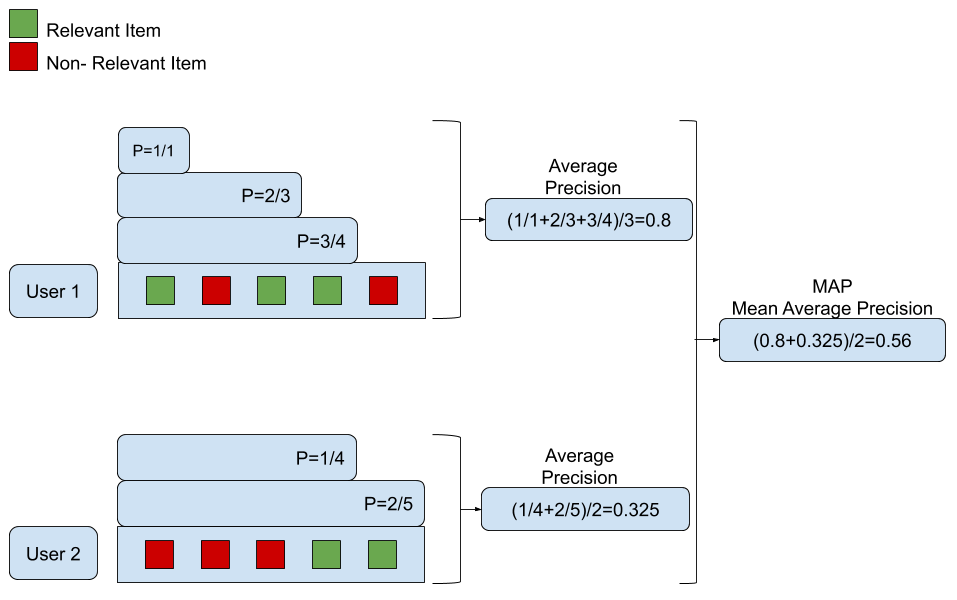
## 2.1 原理

MAP（Mean Average Precision，平均精度均值）是每个主题的平均准确率的平均值，是反映系统在全部相关文档上性能的单值指标。系统检索出来的相关文档越靠前(rank 越高)，MAP就可能越高。如果系统没有返回相关文档，则准确率默认为0。

假设我们有一个二进制相关性数据集。我们想要评估整个推荐项目列表，直到一个特定的截止值n。这个截止值之前使用Precision@N度量。决策支持度指标计算n个推荐中好的推荐的比例。此指标的缺点是，它不认为推荐列表是一个有序列表。P@N将整个列表视为一组条目，并平等对待推荐列表中的所有错误。

目标是在列表的前几个元素中减少错误，而不是在列表的后几个元素中。为此，我们需要一个度量来相应地对误差进行加权。这样做的目的是要在列表的顶部对错误的权重加大。然后，当我们沿着列表中较低的项目往下走时，逐渐减少错误的重要性。

平均预测(AP)度量试图近似这个加权滑动指标。它结合使用连续子列表上的精度，以及这些子列表中召回率的变化。计算如下：



从上图中，我们可以看到平均精度度量是在单个推荐列表，即用户级别。通过此项计算精度意味着细分推荐列表。当我们得到一个相关的项目时，我们就检查一个新的子列表。然后计算当前子列表的精度。我们对每个子列表都这样做，直到我们到达推荐的末尾。现在我们有了一组精度，我们对它们进行平均以获得单个用户的平均精度。然后我们得到所有用户的AP和平均精度。

例如：假设有两个主题，主题1有4个相关网页，主题2有5个相关网页。某系统对于主题1检索出4个相关网页，其rank分别为1, 2, 4, 7；对于主题2检索出3个相关网页，其rank分别为1,3,5。对于主题1，平均准确率为(1/1+2/2+3/4+4/7)/4=0.83。对于主题2，平均准确率为(1/1+2/3+3/5+0+0)/5=0.45。则MAP= (0.83+0.45)/2=0.64。”

## 2.2 MAP优点

* 给出了一个代表精确度—召回率曲线下复杂区域的单一度量。这提供了每个列表的平均精度。
* 处理列表推荐物品的自然排序。这与将检索项视为集合的度量标准形成了对比。
* 这一指标能够给予发生在排序高的推荐名单中的错误更多的权重。相反，它对发生在推荐列表中较深位置的错误的权重较小。这符合在推荐列表的最前面显示尽可能多的相关条目的需要。

## 2.3 MAP缺点

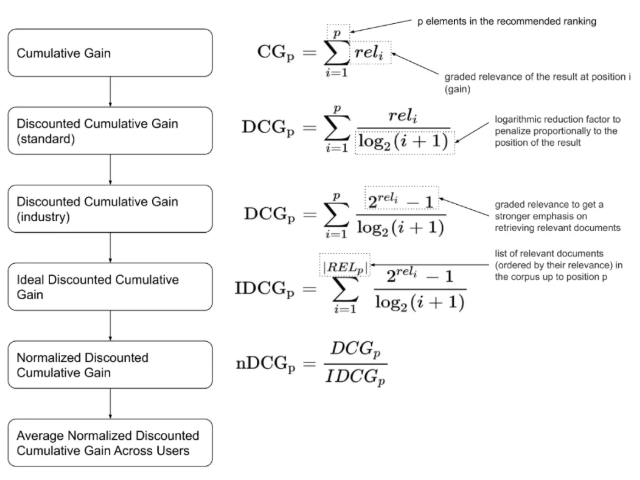
* 这个度量标准适用于二进制(相关/非相关)评级。然而，它不适合细粒度的数字评级。此度量无法从此信息中提取误差度量。
* 对于细粒度的评分，例如从1星到5星的评分，评估首先需要对评分进行阈值，以产生二元相关性。一种选择是只考虑大于4的评级。由于人工阈值的存在，这在评估度量中引入了偏差。此外，我们正在丢弃那些精细的信息。这个信息是在4星和5星之间的差异评级，以及在不相关的项目的信息。1星评级真的和3星评级一样吗？

# 3 标准化折扣累积收益（NDCG）

## 3.1 原理

NDCG（Normalized Discounted cumulative gain），归一化折损累计增益。它能够利用某些文档比其他文档“更”相关这一事实。高度相关的项目应该在中度相关项目之前，中度相关项目应该在非相关项目之前。。DCG的两个思想：

* 高关联度的结果比一般关联度的结果更影响最终的指标得分；
* 有高关联度的结果出现在更靠前的位置的时候，指标会越高



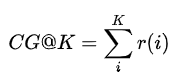
### 3.1.1 增益（Gain）

Gain表示列表中每一个item的相关性分数：



### 3.1.2 累计增益（CG）

　　CG，cumulative gain，是DCG的前身，只考虑到了相关性的关联程度，没有考虑item在排序列表的位置。它是一个搜素结果相关性分数的总和。其实就是对K个item的Gain进行累加：

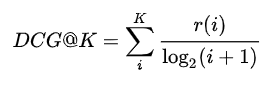


其中，ri代表i这个位置上的相关度。

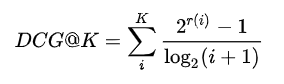
　　举例：假设搜索“篮球”结果，最理想的结果是：B1、B2、 B3。而出现的结果是 B3、B1、B2的话，CG的值是没有变化的，因此需要下面的DCG。

### 3.1.3 折损累计增益（DCG）

对于排序任务，需要增加排序列表中item位置的相对影响。DCG（Discounted Cumulative Gain）是在每一个CG的结果上除以一个折损值，这是为了让排名越靠前的item增益更高，对排名靠后的item进行折损。假设排序越往后，价值越低。到第i个位置的时候，它的价值是1/log2(i+1)，那么第i个结果产生的效益就是ri \* 1/log2(i+1)，所以：

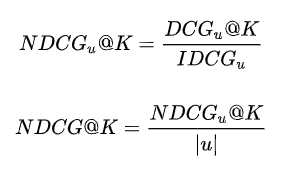


如果相关性分数r(i)只有（0，1）两种取值时，DCG@K有另一种表达。其实就是如果算法返回的排序列表中的item出现在真实交互列表中时，分子加1，否则跳过。

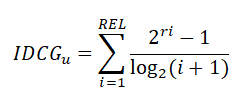


### 3.1.4 归一化折损累计增益（NDCG）

NDCG（Normalized Discounted Cumulative Gain），由于搜索结果随着检索词的不同，返回的数量是不一样，而DCG是一个累加的值，这使得DCG评分在不同检索条件之间没有可比性，我们需要标准化度量，使它在0和1之间，因此需要归一化处理，这里是除以IDCG（Ideal Discounted Cumulative Gain）。然后用每个用户的DCG与IDCG之比作为每次检索结果归一化后的分值，最后对每次检索结果的评分取平均得到最终的分值，即NDCG。



其中，IDCG为理想情况下最大的DCG值。



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。

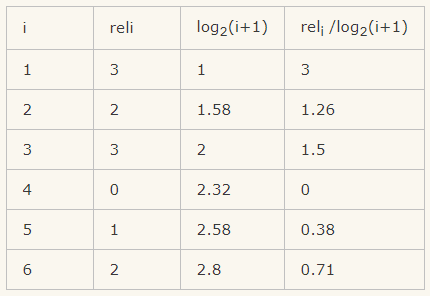
## 3.2 NDCG实例

### 3.2.1 实例1

假设搜索回来的5个结果，其相关性分数分别是3、2、3、0、1、2

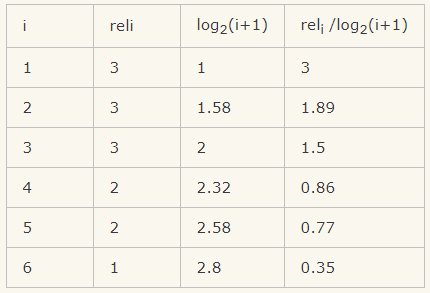
　　那么 CG = 3+2+3+0+1+2

　　可以看到只是对相关的分数进行了一个关联的打分，并没有召回的所在位置对排序结果评分对影响。计算DCG：



所以DCG = 3+1.26+1.5+0+0.38+0.71 = 6.86

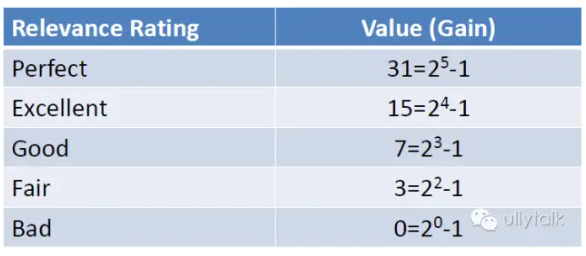
　　接下来归一化，归一化需要先结算IDCG，假如我们实际召回了8个物品，除了上面的6个，还有两个结果，假设第7个相关性为3，第8个相关性为0。那么在理想情况下的相关性分数排序应该是：3、3、3、2、2、1、0、0。计算IDCG@6:



所以IDCG = 3+1.89+1.5+0.86+0.77+0.35 = 8.37

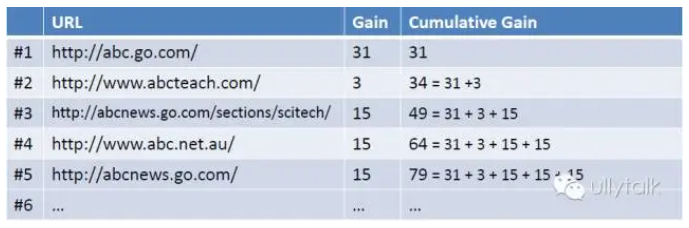
最终 NDCG@6 = 6.86/8.37 = 81.96%

### 3.2.2 实例2

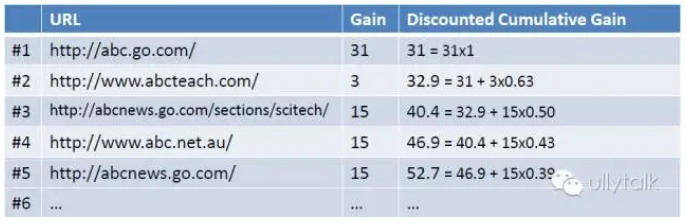


(应该还有r=1那一级，原文档有误，不过这里不影响理解)

例如现在有一个query={abc}，返回下图左列的Ranked List(URL)，当假设用户的选择与排序结果无关(即每一级都等概率被选中)，则生成的累计增益值如下图最右列所示：



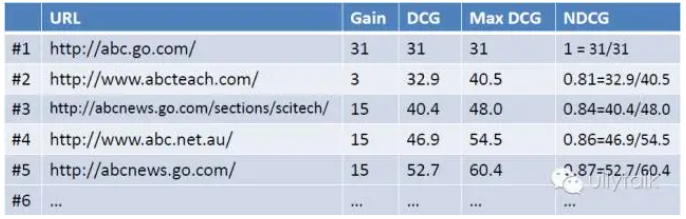
考虑到一般情况下用户会优先点选排在前面的搜索结果，所以应该引入一个折算因子(discounting factor): log(2)/log(1+rank)。这时将获得DCG值(Discounted Cumulative Gain)如下如所示：



最后，为了使不同等级上的搜索结果的得分值容易比较，需要将DCG值归一化的到NDCG值。操作如下图所示，首先计算理想返回结果List的DCG值：



然后用DCG/MaxDCG就得到NDCG值，如下图所示：



## 3.3 NDCG优点

* NDCG的主要优势是它考虑到了分等级的相关性值。当它们在数据集中可用时，NDCG是一个很好的选择。
* 与MAP度量相比，它在评估排名项目的位置方面做得很好。它适用于二元的相关/非相关场景。
* 平滑的对数折现因子有一个很好的理论基础，该工作的作者表明，对于每一对显著不同的排名推荐系统，NDCG度量始终能够确定更好的一个。

## 3.4 NDCG缺点

* NDCG在部分反馈方面有一些问题。当我们有不完整的评级时，就会发生这种情况。这是大多数推荐系统的情况。如果我们有完整的评级，就没有真正的任务去实现！在这种情况下，recsys系统所有者需要决定如何归罪于缺失的评级。将缺少的值设置为0将把它们标记为不相关的项。其他计算值(如用户的平均/中值)也可以帮助解决这个缺点。
* 接下来，用户需要手动处理IDCG等于0的情况。当用户没有相关文档时，就会发生这种情况。这里的一个策略是也将NDCG设置为0。
* 另一个问题是处理NDCG@K。recsys系统返回的排序列表的大小可以小于k。为了处理这个问题，我们可以考虑固定大小的结果集，并用最小分数填充较小的集合。
* NDCG的主要优势在于它考虑到了分级的相关性值。如果你的数据集有正确的形式，并且你正在处理分级相关性，那么NDCG度量就是你的首选指标。

参考

<https://mp.weixin.qq.com/s/0OTxBxriIGUXrJAe-rUyTQ>

<https://mp.weixin.qq.com/s/LyU4hrj67X3BiO5NOkojgQ>

<https://zhuanlan.zhihu.com/p/84206752>

<https://mp.weixin.qq.com/s/cnd9zwdZ68BLVsOae-xp2g>

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