MRR vs MAP vs NDCG：具有排序意义的度量指标的可视化解释及使用场景分析

原创 ronghuaiyang AI公园 7月4日

作者：Moussa Taifi, Ph.D
编译：ronghuaiyang

导读

3种指标，各有优缺点，各有适用场景，分析给你看。

机器学习度量之旅

在不适当的度量指标上报告小的改进是一个众所周知的机器学习陷阱。理解机器学习(ML)指标的优缺点有助于为ML从业者建立个人信誉。这样做是为了避免过早宣布胜利的陷阱。理解用于机器学习(ML)系统的指标很重要。ML实践者需要投资很大的预算将原型从研究转移到生产。中心目标是从预测系统中提取价值。离线度量是推动一种新模式投入生产的关键指标。

在这篇文章中，我们来看看三个排序指标。排序是一项基本任务。它出现在机器学习、推荐系统和信息检索系统中。我最近很高兴地完成了一个优秀的推荐系统专业：明尼苏达大学推荐系统专业。这个专业是一个5门课程的推荐系统课程。我想分享一下我是如何学会评估推荐系统的。特别是当手头的任务是一个排序任务时。

为了不失一般性，大多数推荐系统做两件事。它们要么尝试预测用户对某个物品的评价，要么为每个用户生成一个推荐物品的排序列表。

> 很难想象如何评价一个推荐系统。将**推荐的**项目列表与**相关性**项目列表进行比较是不直观的。传统的任务预测谁死于泰坦尼克号，或者在ImageNet数据集中预测什么品种的狗。它们并不强调排名感知的ML指标，而这些指标是推荐系统的核心。

如果你对此感兴趣，请继续阅读我们探索的评价推荐系统的3个最流行的排名感知指标:

- **MRR**：平均排名的倒数
- **MAP**：平均精度均值
- **NDCG**：标准化折扣累积收益

无排序的度量指标

准确率度量

在处理排序任务时，预测精度和决策支持指标都不高。预测精度指标**包括平均绝对误差(**MAE**)、均方根误差(**RMSE)。这些主要是比较实际的和预测的差距。它们在个人评级预测等级上运作。如果一个用户给一件物品评级为4.5，这些指标告诉我们，如果我们预测的评级为1.2或4.3，我们的预测距离有多远。

决策支持的度量

接下来，**决策支持**指标包括**精度、召回率**和**F1得分**。这些重点是衡量推荐人如何帮助用户做出好的决定。它们帮助用户选择“好的”物品，并避免“坏的”物品。这些类型的度量开始强调对推荐系统来说什么是重要的。如果我们向用户推荐100个物品，最重要的是前5个、10个或20个位置的物品。精确度是选出来的物品中与用户相关的物品的百分比。它的重点是推荐最有用的东西。召回率是推荐系统选择出来的相关物品占所有相关物品的百分比。它的重点是不缺少有用的东西。F1得分是两者的结合。F1调和平均值是一种平衡精度和召回率的方法，得到一个单一的度量。

对于我们的排序任务，这些度量有一个主要的缺点。这些决策支持度量覆盖了整个数据集。它们不是针对“最顶端”的推荐。precision和recall都是关于整个结果集的。为了扩展这些度量，precision和recall通常都有一个上限n。它的形式是**Precision@N**和**Recall@N**。有趣的是，我找不到一个好的来源来描述代表P@N和R@N的调和平均数的**F1@N得分**。我们继续吧。

修改后的Precision@n度量是好的“top-n”的百分比。这包含某种级别的top-n评估。这意味着它将重点放在最受推荐的项目上。然而，它们仍然类似于最初的精度、召回和F1。它们主要是关于如何善于发现事物。我们需要一些指标来强调我们是否善于发现并对事物进行排名。

是时候升级了。让我们看看有排序意识的评估指标是如何发挥作用的。

有排序意义的度量指标

推荐系统有一个非常特别和主要的关注点。他们需要能够把相关的物品放在推荐列表的最前面。最可能的情况是，用户不会通过滚动浏览200个条目来找到他们最喜欢的伯爵茶品牌。我们需要基于排名的指标来选择推荐物品，以达到以下两个主要目标：

1. 推荐系统把推荐的物品放在哪里？
2. 推荐系统建模**相对偏好**的能力如何？

这就是以下指标可以帮助的地方：

> MRR: Mean Reciprocal Rank
>
> MAP: Mean Average Precision
>
> NDCG: Normalized Discounted Cumulative Gain

上述3个度量标准来自于两个度量家族。第一种度量包括基于二进制相关性的度量。这些度量标准关心的是一个物品在二进制意义上是否是好的。第二个系列包含**基于应用的度量**。它们通过度量绝对或相对的好来扩展好/坏的感觉。让我们在下一节中描述每个度量的特点。

MRR: Mean Reciprocal Rank

这是三者中最简单的度量。它试图度量“第一个相关的物品在哪里？”它与二元相关性度量族密切相关。算法如下：

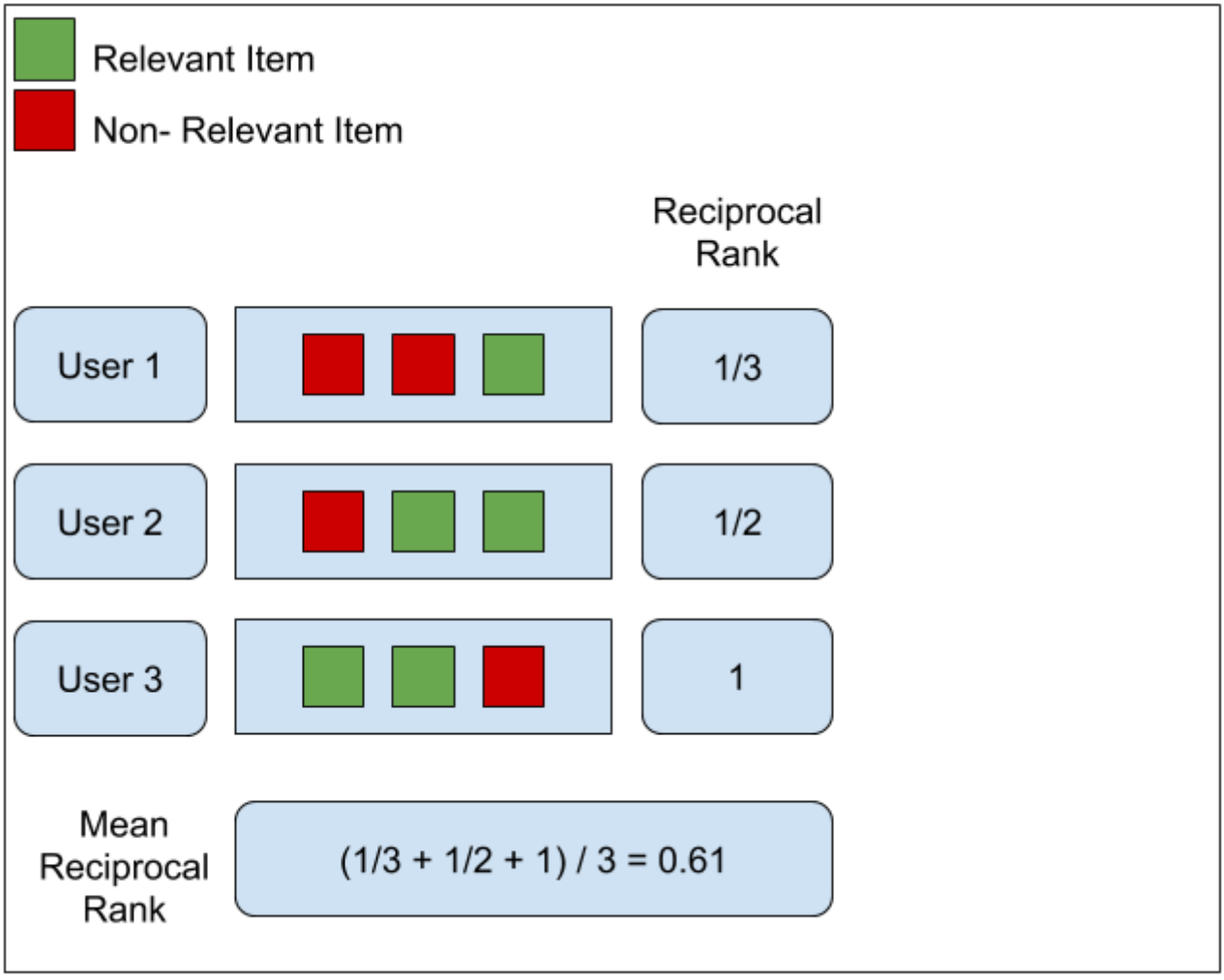
For each user u:

- Generate list of recommendations
- Find rank k_u of its first relevant recommendation (the first rec has rank 1)
- Compute reciprocal rank $\frac{1}{k_u}$

Overall algorithm performance is mean recip. rank:

$$\mathrm{MRR}(O, U) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k_u}$$

假设我们有以下三个针对三个用户的推荐列表。我们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。



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。

MRR的缺点

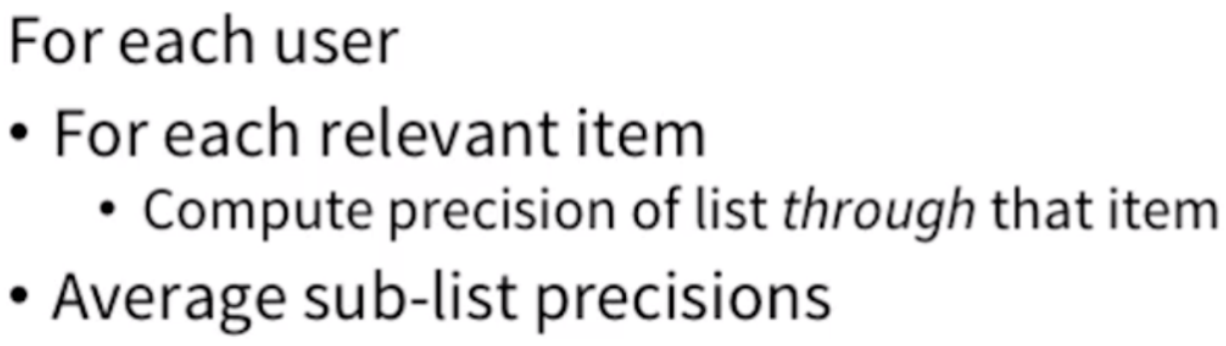
- MRR指标不评估推荐项目列表的其余部分。它只关注列表中的第一个项目。
- 它给出一个只有一个相关物品的列表。如果这是评估的目标，那找个度量指标是可以的。
- 对于想要浏览相关物品列表的用户来说，这可能不是一个好的评估指标。用户的目标可能是比较多个相关物品。

MAP: Average Precision and Mean Average Precision

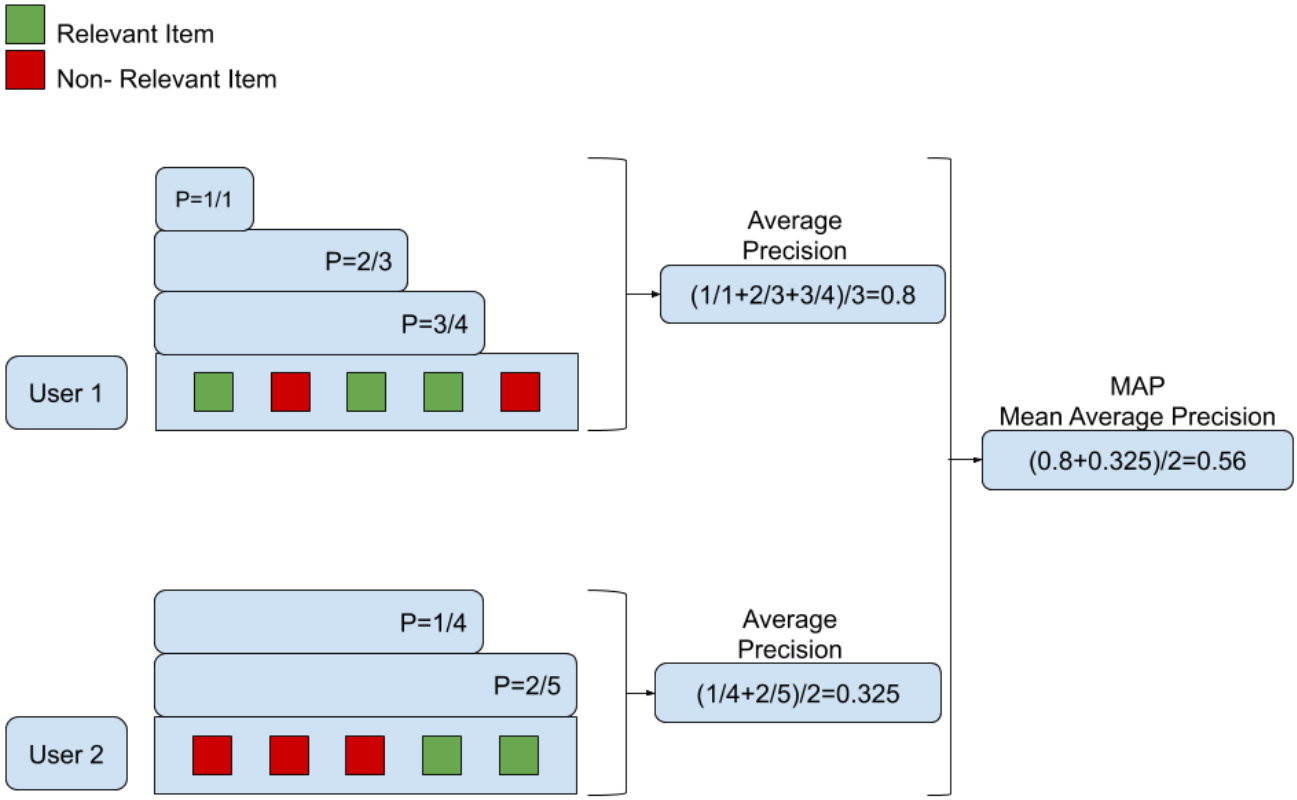
接下来是MAP度量。假设我们有一个二进制相关性数据集。我们想要评估整个推荐项目列表，直到一个特定的截止值n。这个截止值之前使用Precision@N度量。决策支持度指标计算n个推荐中好的推荐的比例。此指标的缺点是，它不认为推荐列表是一个有序列表。P@N将整个列表视为一组条目，并平等对待推荐列表中的所有错误。

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：

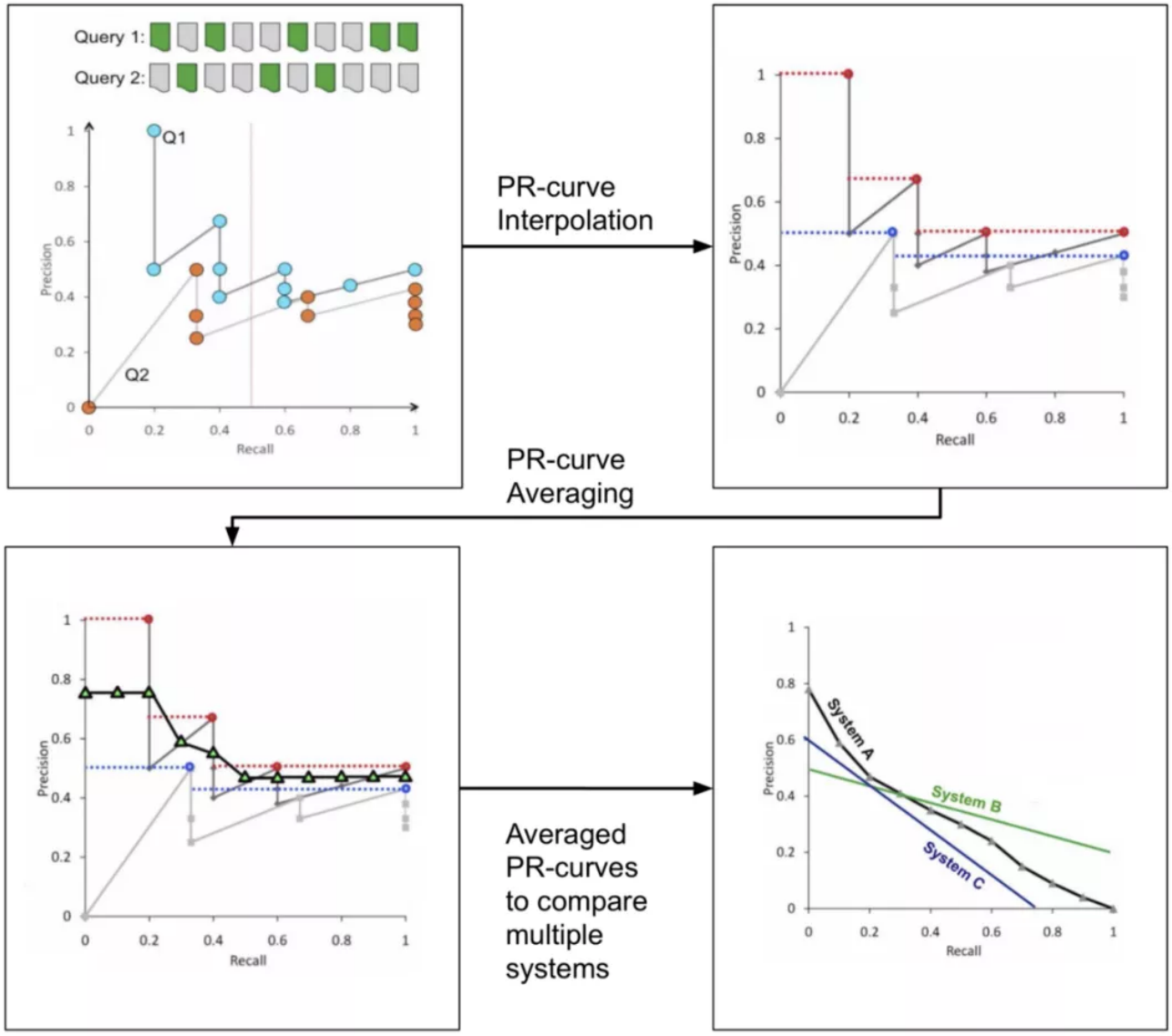


这里有一个图来帮助可视化这个过程：



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。

这主要是AP度量的原始目标的近似值。AP度量表示精确度召回率率曲线下的面积。通过计算召回率作为召回值的函数，得到了精确度-召回率曲线。整个过程就是为每个用户推荐列表生成PR曲线。然后生成插值后的PR曲线，并对插值后的PR曲线求平均。这是视觉上的过程：



通过PR曲线下的面积进行MAP的度量

为了比较两种系统，我们需要PR曲线下尽可能大的区域。在上面的例子中，我们比较了系统A, B和C。我们注意到系统A比系统C在所有级别的召回上都要好。但是，A系统和B系统相交的地方是B系统在较高的召回水平上表现更好。这个场景的问题是很难确定哪个系统总体上做得更好。绘图比单一的指标更难解释。这就是为什么研究人员提出了一个单一的度量来近似平均精确度(即精确度 —— 召回率曲线下的面积)。

Area under the PR-Curve

$$\text{AveP} = \int_0^1 p(r)dr$$

Finite sum approximation over the every ranked recommendations

$$\text{AveP} = \sum_{k=1}^{n} P(k)\Delta r(k)$$

number of retrieved items

change in recall from items k to k-1

precision at cut-off k in the list

rank in the sequence of retrieved items

Simplified to

$$\text{AveP} = \frac{\sum_{k=1}^{n}(P(k) \times \text{rel}(k))}{\text{number of relevant documents}}$$

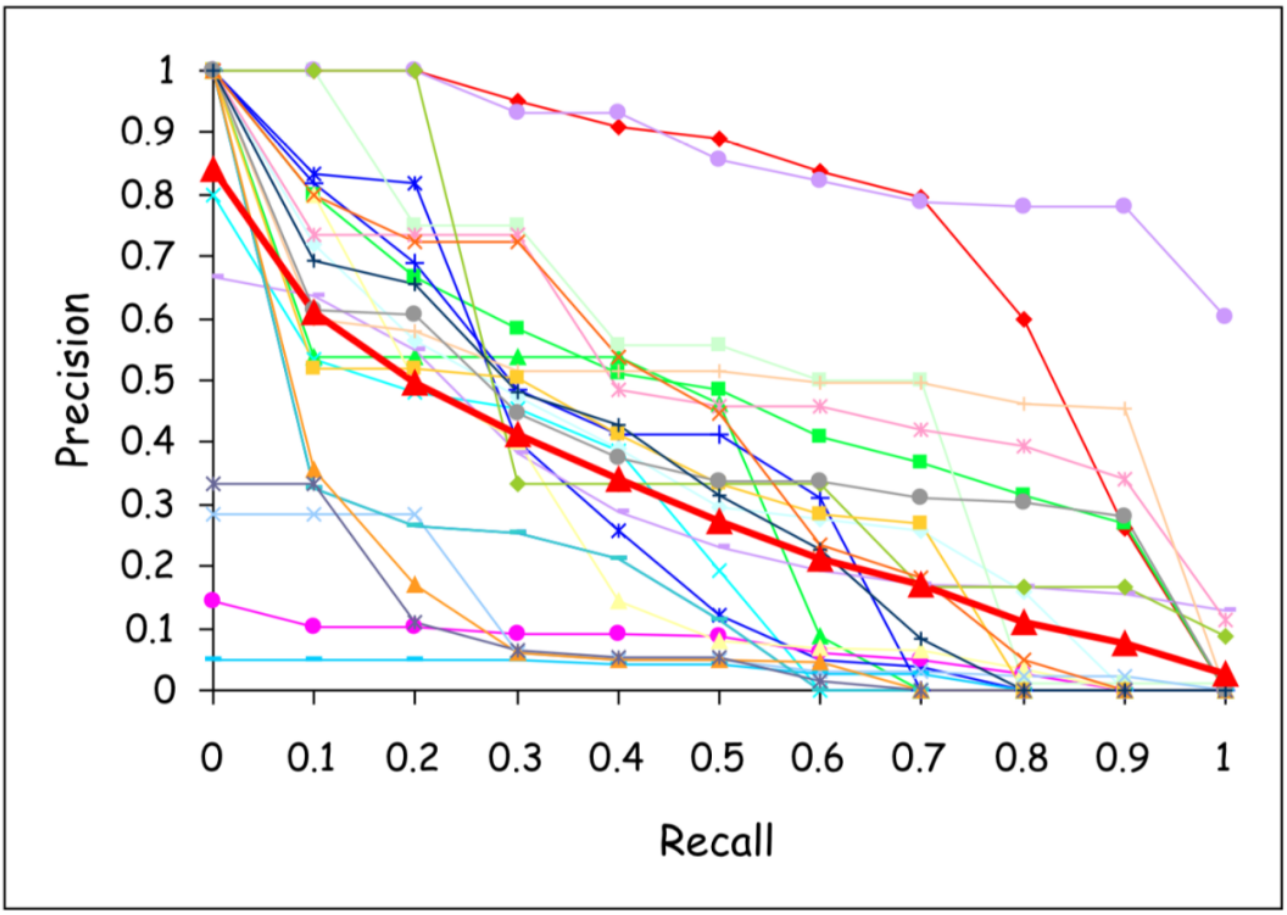
indicator function equaling 1 if the item at rank k is a relevant item

In code to return a vector of average precision scores for every k-th element. true_positive is an ordered list of relevancy of recommended items

```
tp_cumsum = np.cumsum(true_positive)
val_counter = np.cumsum(np.ones(len(true_positive)))
return np.cumsum(tp_cumsum * true_positive / val_counter) / tp_cumsum
```

离散MAP度量的推导

最后一点是要意识到我们实际上是在求均值。这意味着对多个用户的噪声信号进行平均。下面是一个带有噪声的图，在许多用户中是常见的。在解释MAP分数时，记住这一点是很有用的。在下图中，我们可以看到鲜亮的红线是pr曲线的平均值。下面图中的其他曲线是针对N个用户列表中的每个用户的。实际上，查询的有效性可能存在很大差异。MAP的平均值无疑会对报告的性能产生影响。这样的样本曲线可以帮助评价MAP度量的质量。



演示了在许多用户中MAP度量的影响。

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。

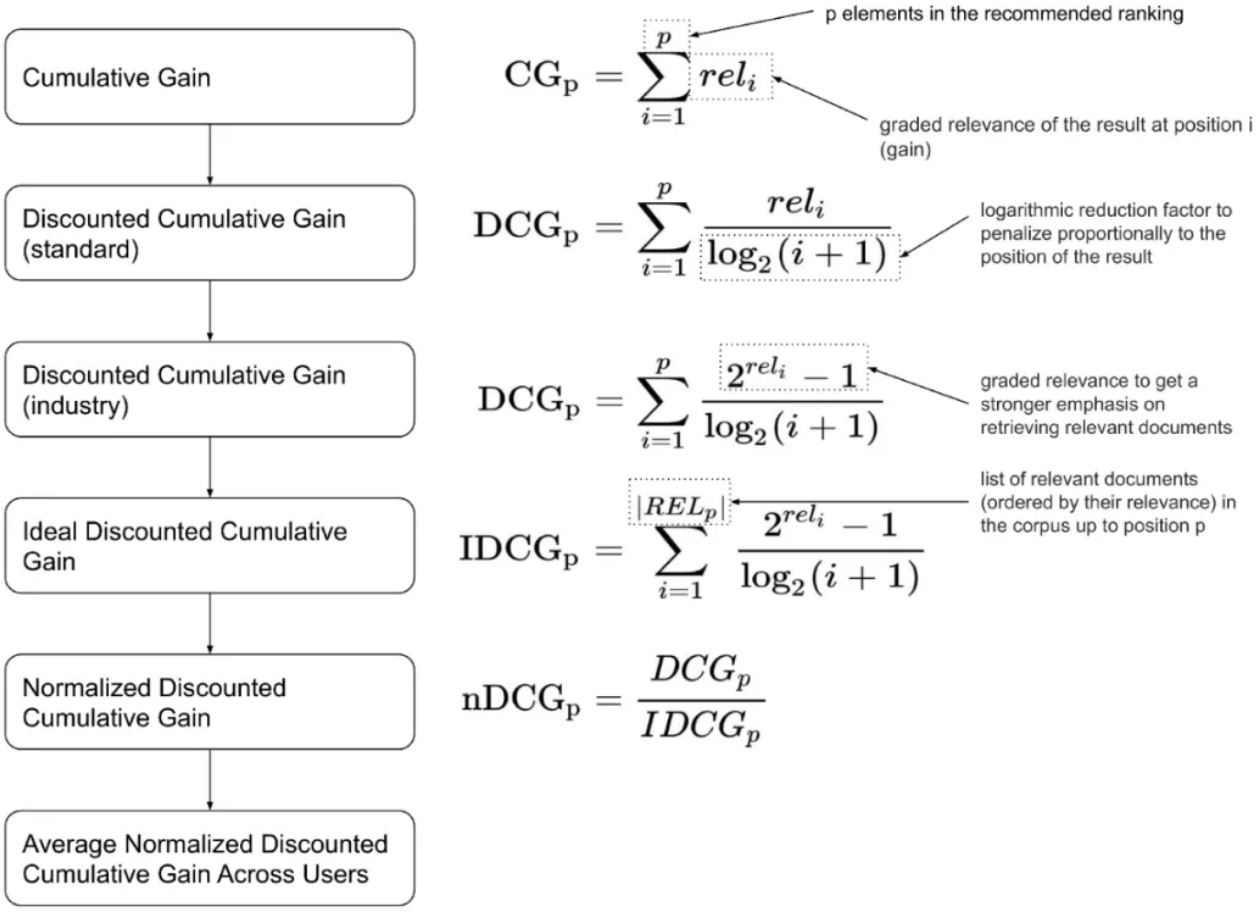
MAP缺点

- 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？

为了解决这些问题，recsys社区提出了另一个更近期的度量标准。这个度量考虑了评级中包含的细粒度信息。让我们看一看NDCG度量。

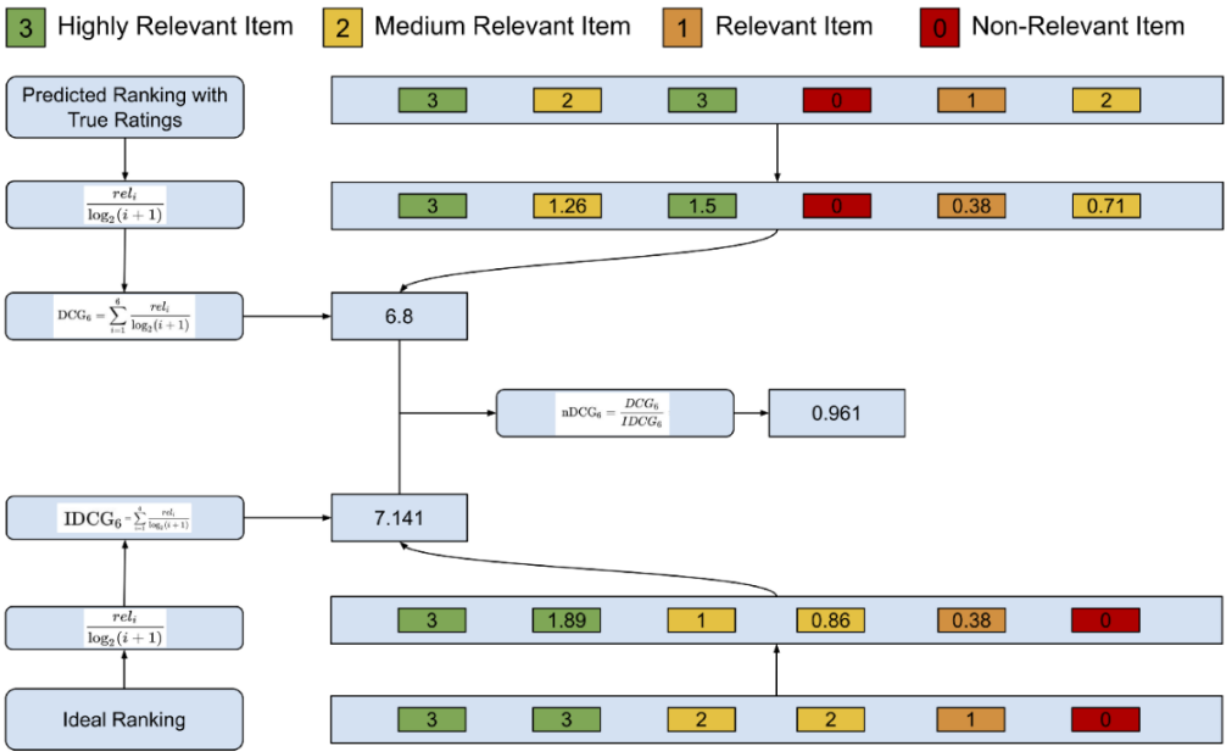
Normalized Discounted Cumulative Gain

MAP度量的目标与NDCG度量的目标相似。它们都重视将高度相关的文件排在推荐列表的前列。然而，NDCG进一步调整了推荐列表评估。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。我提供以下图表，它显示了阶段计算NDCG的步骤：



在NDCG之前我们有cumulative gain CG。这是一种基本的方法来积累等级相关度。这个度量不考虑元素在排序列表中的位置。对于排序任务，我们需要增加排序列表中元素位置的相对影响。standard Discounted Cumulative Gain(DCG)增加了一个对数衰减因子，以按比例惩罚项目的位置相关分数。此外，在工业应用中，为了强调检索相关文档，相关性分数得到提升是很常见的。这出现在industry DCG公式中。

我们在处理动态系统。用户将得到数量可变的相关项目推荐。这使得DCG测量在用户之间没有可比性。我们需要标准化度量，使它在0和1之间。为此，我们确定用户的理想排名。然后用该排序作为Ideal Discounted Cumulative Gain IDCG。这提供了一个很好的归一化因子。它有助于计算 Normalized Discounted Cumulative Gain。因为这是一个针对每个用户的度量，所以我们需要为测试集中的所有用户计算这个度量。然后，这个平均值用于比较recsys系统之间的差异。为了可视化这个过程，我们在下面的图中计算单个用户的预测和理想排名。



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。

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。

正如我所说的，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。