案例|推荐系统的评估指标

第四范式...
已认证的官方帐号
关注他

推荐系统能够为用户提供个性化体验，现在基本上各大电商平台、资讯平台都会用推荐系统为自家评价下的用户提供千人千面的服务。平均精度均值（Mean Average Precision，MAP）便是评估推荐系统性能的度量标准之一。

但是，使用其他诊断指标和可视化工具可以让模型评估更加深入，甚至还会带来一些其他启发。本文探讨了召回率、覆盖率、个性化和表内相似性，并使用这些指标来比较三个简单的推荐系统。

Movielens数据集

这篇文章中的例子使用的数据是Movielens 20m数据集。这些数据包含用户对电影的评分以及电影类型的标记。 （为了延长训练时间，该数据被下采样，评分仅包括给超过1000部电影打过分的用户的评分，以及3星及其以上的评分。）

userId	movieId	rating
156	1	5.0
156	2	5.0
156	4	3.0

用户电影评级的示例

模型

本文测试并比较了三种不同的推荐系统：

1.随机推荐（随机为每位用户推荐10部电影）

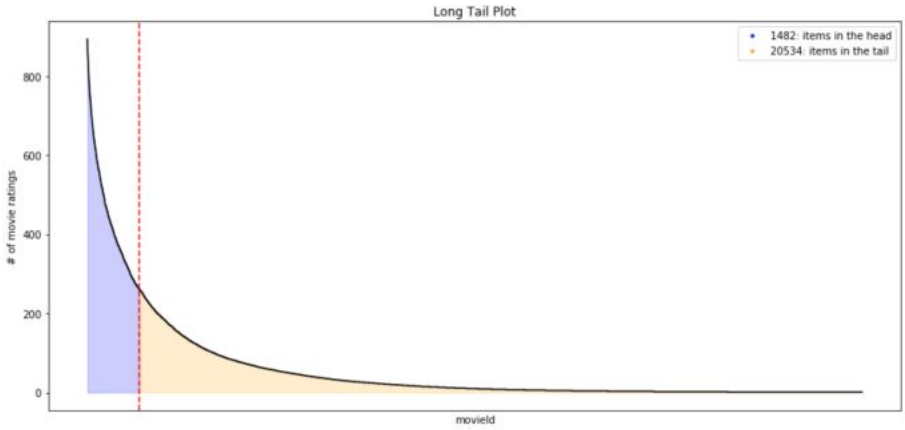
2.根据流行度推荐（向每位用户推荐最受欢迎的10部电影）

3.协同过滤器（使用SVD的矩阵分解方法）

接下来就让我们深入了解这些指标和诊断图，并比较这些模型！

长尾图

长尾图用于挖掘用户-项交互数据中的流行度模式，例如点击次数、评分或购买行为等。通常，只有一小部分项目具有大量的交互，我们称之为“头部”；而大多数项目都集中在“长尾”中，它们只占交互的一小部分。



长尾图 （Movielens 20m评级数据样本）

在训练数据中会对许多热门项目进行多方观察，因此，推荐系统想要准确预测这些项目并不难。在电影数据集中，最受欢迎的电影是大片和经典老片。这些电影已为大多数用户所熟知，推荐这些电影，对用户来说可能并非是个性化推荐，也可能无法帮助用户发现其他新的电影。相关推荐被定义为用户在测试数据时给予正面评价的项目的推荐。这里的指标用来评估推荐系统的相关性和实用性。

MAP和MAR

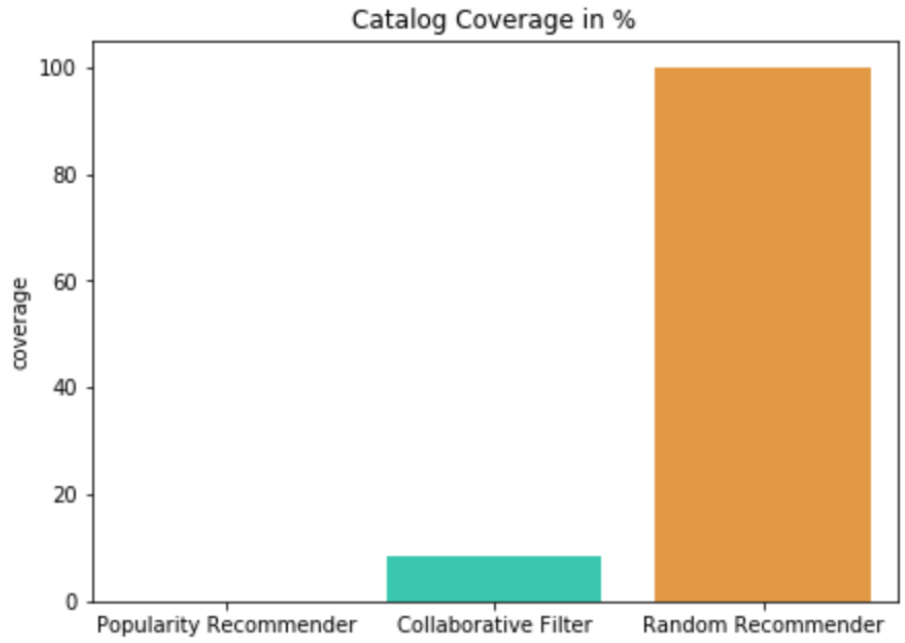
推荐系统会为测试集中的每个用户生成推荐的有序列表。平均精度均值（MAP）可以让开发者深入了解推荐项目列表的相关性，而召回率可以让开发者深入了解推荐系统的调试性能，如调试用户给予正向评价的所有项目。MAP和MAR的详细描述如下：

Mean Average Precision (MAP) For Recommender Systems
sdsawtelle.github.io

覆盖率

覆盖率是指模型能够在测试集上推荐的项目占训练数据的百分比。在此示例中，受欢迎度推荐的覆盖率仅为0.05%，它只推荐了10件物品。随机推荐器的覆盖率接近100%。出乎意料的是，协同过滤只能推荐其训练的项目的8.42%。

三个推荐系统的覆盖率比较：



个性化

个性化是评估模型是否向不同用户推荐相同项目的方法。用户的推荐列表之间存在差异（1-余弦相似性）。下边的例子能很好地说明如何计算个性化程度。

3个不同用户的推荐项目示例列表：

```
example_predictions = [
    ['A', 'B', 'C', 'D'],
    ['A', 'B', 'C', 'X'],
    ['A', 'B', 'C', 'Z']
]
```

首先，每个用户的推荐项目会被表示为二进制指示符变量（1：向用户推荐该项目.0：不向用户推荐该项目）。

	A	C	B	D	X	Z
0	1	1	1	1	0	0
1	1	1	1	0	1	0
2	1	1	1	0	0	1

然后，跨所有用户的推荐向量计算余弦相似度矩阵。

```
[1.   , 0.75, 0.75]
[0.75, 1.   , 0.75]
[0.75, 0.75, 1.   ]
```

最后，计算余弦矩阵的上三角的平均值。个性化是1-平均余弦相似度。

$$personalization = 1-0.75 = 0.25$$

高个性化分数表示用户的推荐不同，这也意味着该模型为每一位用户提供个性化体验。

列表内相似性是推荐列表中所有项目的平均余弦相似度。该计算使用推荐项目（例如电影类型）的特征来计算相似度。该计算方法可以通过以下示例说明。

针对3个不同用户的电影ID的推荐示例：

```
example_predictions = [
    [3, 7, 5, 9],
    [9, 6, 12, 623],
    [7, 894, 6, 623]
]
```

这些电影类型特征用于计算推荐给用户的所有项目之间的余弦相似度。该矩阵显示了向用户1推荐的所有电影的特征。

movieId	Action	Comedy	Romance
3	0	1	0
7	0	1	0
5	0	1	0
9	1	0	0

我们可以为每个用户计算表内相似性，并对测试集中的所有用户求平均值，从而得到对模型的表内相似性的估计。

```
recmetrics.intra_list_similarity(example_predictions, feature_df)
```

```
0.27777777777777773
```

如果推荐系统向每一个用户推荐非常相似的项目列表（如用户仅接收浪漫电影的推荐），那么列表内相似性将很高。

使用正确的训练数据

我们可以对训练数据进行如下操作，从而快速改进推荐系统：

1.从培训数据中删除热门项目 （这一点适用于用户可以自行找到这些项目，以及发现这些项目不具备实用性的情况）。

2.按照用户的值来放大项目评级，例如平均交易值。这样做有助于模型推荐能够带来忠诚度或高价值客户的项目。

结论

一个好的推荐系统能够生成兼具实用性和相关性的推荐结果。

使用多个评估指标来评估模型，能够更加全面地衡量一个推荐系统的性能。

原文链接：Evaluation Metrics for Recommender Systems