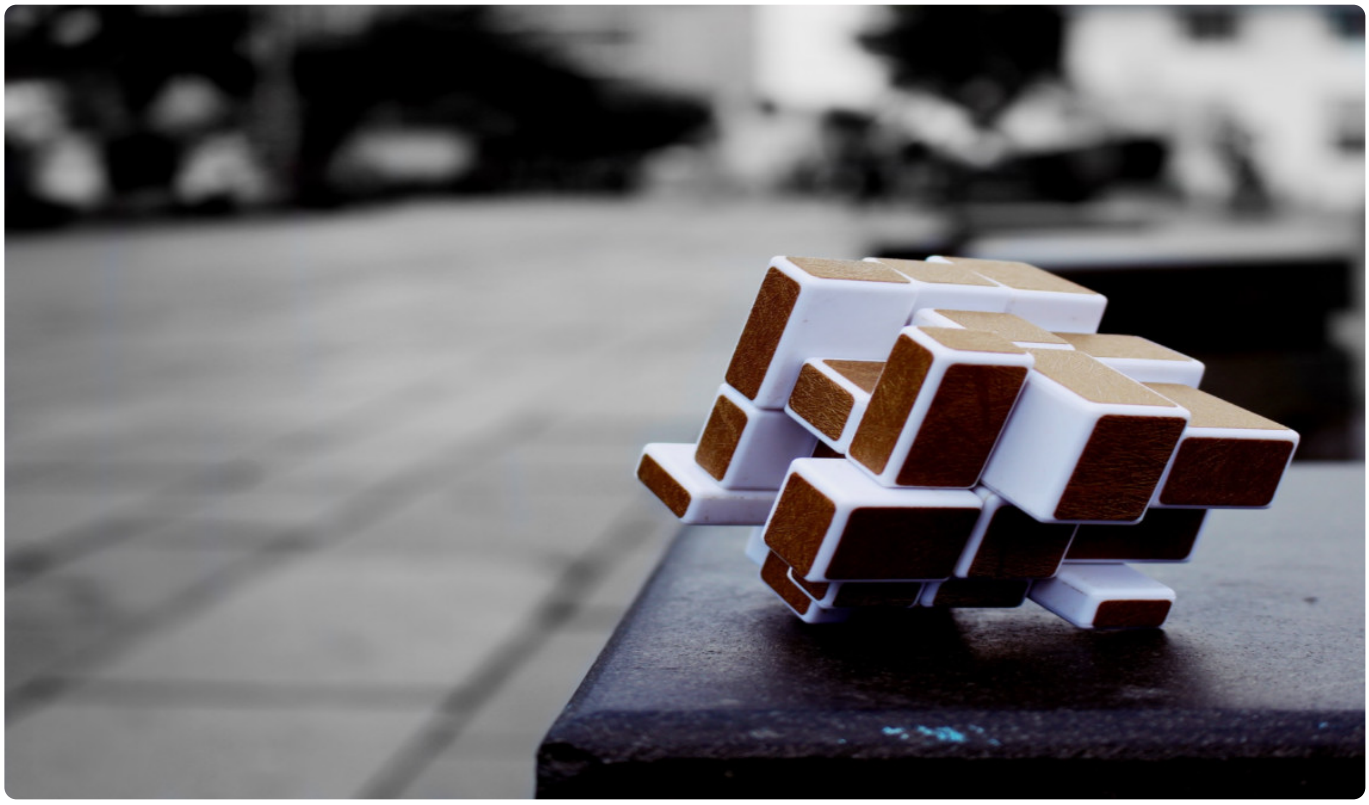
123 | 数据科学家必备套路之二：推荐套路

2018-07-16 洪亮劼

AI技术内参 进入课程 >



讲述：初明明
时长 07:34 大小 3.47M

在上一期的分享里我们讨论了做搜索产品的套路，给你介绍了多轮打分、高频和长尾以及三大模型套路。你有没有感受到这些高于某一个具体模型的套路的重要性呢？

今天，我们来看看推荐的一些套路。

多轮打分套路

上一篇我们提到，想要构建一个搜索引擎，应该立刻想到基于多轮打分的架构，有这个意识就是一个基本套路。

其实这个套路对于推荐，也是适用的。

把推荐问题构建成一个多轮打分的“类搜索”问题，其实是推荐在工业界应用的一个非常重要的套路。

这个思路的好处是把搜索和推荐问题给归一化了。也就是说，我们可以依靠同样一套软件架构来解决两大类相似的问题。**搜索是有关键词的推荐，而推荐则是无关键词的搜索。**虽然这是一种相对比较简化的看待这两种问题的方式，但是统一的架构在工程上面可以带来非常多的好处，比如重复构建相似的特征工程的流水线，以及更重要的如何优化索引等工程，这些都可以很快地应用在搜索和推荐这两个重要的场景上。

当然，在工程以外还有其他好处。在学术界，关于推荐系统搭建的方法往往是一种独立的模型，然后搜索系统又是另外一种独立的模型。这些模型之间缺乏能够系统性联系起来的纽带。**把推荐问题看成是多轮打分的搜索问题**之后，我们就找到了一种简单又自然的方法，能够把很多不同类型的推荐模型给整合到一起。

比如，很多之前我们介绍过的推荐模型就可以担任第一轮打分，也就是我们常说的“**候选集选择**”（Candidate Selection）这一组件的角色。像协同过滤模型，就可以是我们为每一个用户或者每一个物品产生最终推荐物品的一个候选集合。

在搜索里，我们是利用索引以及简单的检索方法，从海量的文档中找到几百或者几千个初步相关的文档，然后再根据第二轮的复杂模型来重排序。那么在推荐里，我们其实就可以利用各种不同的协同过滤、矩阵分解等模型来达到第一步的筛选功能。

而对于第二轮打分，我们就完全可以依赖基于特性的排序学习模型来学习推荐的结果。这种方式其实是极大地利用各种搜索算法，特别是排序学习的进步，来提升推荐的效果。

是否把推荐问题看成是多轮打分的搜索问题，是区别工业界和学术界推荐模型的一个重要标志。

高频用户和低频用户套路

既然我们提到了把推荐问题看成是某种意义上的搜索问题，那么，根据用户行为的频率来进行不同的推荐策略，其实就是一个顺理成章的套路了。

这个套路的思路和搜索类似。对于高频用户而言，我们有足够多的数据，所以往往可以学习到一个比较好的模型。而且，对于真正的高频用户来说，提高推荐的质量往往需要个性化，也就是说，我们需要更多地利用这些高频用户他们自己的数据，来提供推荐结果。

一般来说，针对高频用户的个性化推荐有两种比较常见的方法。

一种方法就是**构造更多的高频用户的特性。**比如，有一个用户点击了某一个物品的信息，或者这个用户购买了某一个物品，这些特性都有助于我们的模型学习到关于这个用户的具体喜好。

另外一种比较常见的方法是**为这些高频用户单独构建模型。**这个方法其实主要是针对第二轮打分的模型而言的。一般来说，一个比较简单直观的方法是把所有用户的数据收集起来，然后训练一个全局的第二轮打分模型。这样做的好处当然是可以利用所有的数据，并且学习出来的模型往往也比较稳定。但是，一个全局的模型往往并不能为某一个用户提供最优的推荐结果，这一点其实很容易理解，因为一个全局的模型往往是某种“平均结果”。所以，我们可以根据用户的数据来为这些高频用户“定制模型”。

说了针对高频用户的一些思路以后，我们来看看针对低频用户的一些套路。

当我们需要为低频用户进行推荐的时候，因为数据缺乏的关系，这时候的选择就不太多了。一个普遍使用的方法，是**对低频用户进行分组。**这种分组一般来说是根据用户的人口信息，例如年龄、性别和地理位置。分组之后，我们把这些组别中的用户信息整合起来，统一建立这些组别的模型。

还有一个比较普遍方法，是**给低频用户推荐流行的信息。**这里的假设是，流行信息之所以是流行的，就是因为这些信息本身可能就有较高的点击率、驻留时间和购买率，因此在不清楚这些低频用户喜好的情况下，推荐这些内容其实是相对比较合理、也是保险的。

批量和实时套路

这个“批量和实时”套路其实和多轮打分以及高频、低频用户都有一些关联，但是有时说的是不太一样的事情。

在设计推荐系统架构的时候，我们刚才讲了多轮打分的思路，那是不是每一个用户到我们的网站或者服务时，系统都需要从第一轮开始一直到最后一轮，完全重新生成一个用户的所有推荐结果？

其实，我们可以这么想一想，如果一些用户，特别是低频用户，每周仅仅光顾几次我们的网站或服务，甚至每个月才光顾一次，我们并不需要针对这些用户来实时更新推荐结果，而可以按照一定的频率，例如每天一次或者每周一次提前生成好所有这些用户的推荐结果，然后

存储到某一个地方。等用户访问网站时，我们就可以直接从存储中调出已经生成好的推荐结果。

其实这个思路不仅仅用于低频用户，高频用户也可以采用这样的方式。不过，更新推荐结果的频率可能就不是每天或者每周，而应该是每几个小时、每几十分钟甚至是更短的时间。

对于很多应用来说，推荐的结果其实并不需要是实时的。即便是在很多看似需要实时的应用上，我们依然**可以用很多的批量计算来达到推荐的目的。**

举个例子，在很多移动场景中，我们可以为一个用户生成一个基本的推荐结果，一两百个物品，然后从服务器端推送到用户的手机上。当用户在手机上产生了新的行为之后，我们可以根据这些行为对用户已经在手机上的这个集合进行模型的微调，然后重新排序。这里用户看到的可能是感觉上已经有更新的推荐结果，但**这种实时的效果其实是建立在批量预处理上的。**

能够理解什么时候需要利用批量的计算结果，什么时候需要实时的计算结果，是处理好推荐问题的一个关键套路。

总结

今天我为你介绍了做推荐产品的几个套路。

一起来回顾下要点：第一，把推荐问题看成一个多轮打分的“类搜索”问题，是推荐在工业界应用的一个重要套路；第二，对高频用户进行个性化推荐有两种常用的思路，包括构造更多的特性和定制建模；针对低频用户的推荐套路也有两个，一个是分组一个是推荐流行的信息；第三，我们聊了批量处理和实时处理的套路，关键是判断在什么场景下使用哪种套路。

最后，给你留一个思考题，从多轮打分系统的架构看，推荐和搜索又有哪些区别需要注意呢？

欢迎你给我留言，和我一起讨论。

上一篇　122 | 数据科学家必备套路之一：搜索套路

下一篇　124 | 数据科学家必备套路之三：广告套路

精选留言 (1)

写留言

范深
2018-07-26

多轮打分系统中，推荐和搜索在召回阶段的目标应该是不一样的。搜索需要侧重query相关性，推荐追求转化率即可。