078 | 现代推荐架构剖析之一：基于线下离线计算的推荐架构

2018-04-02 洪亮劼

AI技术内参 进入课程 >

讲述：初明明
时长 07:06 大小 3.25M

上周，我们讨论了推荐系统的评测，聊了推荐系统的线下评测、线上评测和无偏差估计。至此，我们已经聊了推荐系统的一些基本技术和评测体系，相信你已对推荐系统有了一个基本的认识。

那么，到底该如何搭建一个工业级的推荐系统呢？这周，我们就来谈一谈**现代推荐系统的架构体系**，帮助你从宏观上对推荐系统的构建有一个更加完整的认识。

今天，我们先来看一看，**基于线下离线计算的推荐架构**。

推荐架构需要解决的问题

在讨论任何一种架构之前，我们首先来看一下这个架构需要解决什么样的问题。然后在这些问题的指引下，我们就可以来分析不同架构在解决这些问题上的优劣。

那么，对于一个推荐架构来说，我们需要解决什么样的问题呢？

首先，从大的角度来说，一个用户来到我们的网站或者服务，**推荐系统最重要的一个任务就是，能够在一两百毫秒内给用户提供当前的推荐结果。**也就是说，从用户的角度来看，**推荐结果的呈现必须是实时的。**这一条就是把工业级应用和学术模型区分开的一个重要指标。

在过去十多年里，学术界的推荐系统，或者是 Kaggle 竞赛的推荐系统，往往是一个使用了很多不同模型的**集成模型**（Ensemble Model），这种方式虽然在比赛和论文发表中能够取得较高的精度，但是在现实的系统中，如果不加修改直接使用，必然无法在规定的时间内，也就是一两百毫秒内产生所有的推荐结果。同样的，很多非常复杂的深度学习模型，也无法在规定的时间内产生所有的推荐结果。由此可见，很多推荐架构的核心就是在解决这些问题。

其次，**推荐系统架构需要对用户和系统的交互结果做出响应。**什么意思呢？如果用户看了推荐结果，并且点击了一些推荐结果，或者明确表达了对推荐结果的不喜爱，那么推荐模块需要对这些互动的结果给予回馈。试想，如果一个用户已经明确表示了对某些结果的不喜欢，然后在下一次访问的时候，用户又看到同样的推荐，那一定是一个非常不好的体验。

最后，**推荐系统架构需要考虑用户群体的覆盖率的问题。**如果一个系统架构只能为部分用户服务，这想必无法真正做到对一个网站或者服务产生影响力。因此，在模型以及其他的技术选择上面，如何能够做到“为更广阔的用户群体服务”，就是一个非常关键的因素。

基于线下离线计算的架构

刚才我们简单讨论了一个现代推荐系统架构需要满足的一些需求。那么，在这些需求的驱动下，一种简单易行的架构就诞生了，那就是“**基于线下离线计算的架构**”。

什么叫基于线下离线计算的架构呢？

试想一下，我们有一个推荐模块，是在一个网站首页为用户推荐新闻。现在假设，我们有 M 个用户，N 个新闻文章。M 的数量级可能是几千万，N 的数量级可能是几百万。那么，理想状态下，需要用我们的模型，对每一个用户，以及每一个新闻进行打分。具体地，对于

某一个用户来说，当这个用户访问网站的那一瞬间，我们需要对几百万的新闻进行打分，并且在一两百毫秒内返回结果，这很有可能是不现实的。

既然我们无法**实时**对所有的新闻打分，那么，退一步讲，我们能不能事先把这些打分的工作都做了，然后当用户来到网站的时候，我们仅仅是显示结果呢？答案是，可以的，并且**这就是线下离线计算的核心思想。**

通俗地说，**线下离线计算的一个主要想法**就是：把计算中复杂的步骤尽量提前做好，然后当用户来到网站需要呈现结果的时候，我们要么已经完成了所有的计算，要么还剩非常少的步骤，可以在很快的时间内，也就是所说的一两百毫秒内完成剩下的计算。

回到我们刚才的新闻推荐的例子。我们可以把针对每一个用户的所有新闻的打分，在线下都提前计算好，然后存放在一个数据库或者数据存储的地方，当用户来到网站的时候，我们只需要展示已经完全计算好的推荐结果。

完全线下离线计算的最大好处就是，当用户来临的时候，基本没有任何的计算成本。系统唯一需要做的就是从一个数据存储的地方直接取得当前的推荐结果。

也就是说，线下离线计算的最大好处，就是解决我们刚才说的在规定的时间内计算出推荐结果的需求。然而，线下离线计算对其他两个需求则无法很好地处理。

第一，因为我们是完全提前计算好了所有的结果，并且存储在数据库中。那么，假设用户和推荐结果进行了交互，希望更新推荐结果，离线计算的模式就无法支持这样的操作，或者是非常困难。

我们可以试想一下，如果一个用户不喜欢某一个新闻推荐结果，那么在当前的框架下，我们应该如何应对呢？首先，我们需要启用线下的计算流程，重新计算这个用户所有的推荐结果，然后把这个推荐结果存储到刚才说的数据库里，这样用户下一次来到网站的时候，就会看到更新的结果了。

因为刚才我们已经假设模型的复杂度导致无法很快地进行运算，因此，这个更新的流程可能会比较耗时。同时，这只是一个用户的情况，如果我们要针对大量用户进行这样的处理，那最省力的就是隔一段时间，比如说几个小时就针对那些和系统有交互的用户重新计算一次结果，然后再把更新的结果存入数据库。很明显，在这几个小时的间隙里，用户看到的依然是旧的推荐结果。

第二，完全提前计算好所有结果的情况下，针对新的用户，新的新闻文章就无法进行推荐了。针对这些新用户和新文章来说，完全离线计算这种架构就有一个致命的缺陷。当然，我们也可以依照刚才的思路，也就是说隔一段时间，比如几个小时，就针对当前所有用户和所有新闻，重新计算结果，然后把结果存放到数据库中，但是很明显，这也会导致在这个间歇期内，我们无法对新用户和新文章进行推荐。

完全离线计算的推荐架构适用于一些简单的场景和一些应用的初期架构。很明显，在复杂的网站需求下，单靠提前把所有结果都计算好是不能满足动态的用户需求的。

然而，理解离线计算的需求对于构建复杂架构很有帮助。**我们在设计一个更加复杂的架构时，依然会依靠很多离线计算，用线下时间来换取线上时间。这个思路是现代推荐系统架构中非常重要的一个想法。**

小结

今天我为你讲了一种简单的现代推荐系统的构建思路，那就是基于线下离线计算的推荐架构。

一起来回顾下要点：第一，我们聊了聊推荐架构的需求；第二，我们介绍了什么是离线计算架构，以及这种架构的优缺点是什么。

最后，给你留一个思考题，如果我们的用户数量和物品数量实在是太大，线下计算无法满足每天全部更新一次推荐，这种情况下，我们又该怎么办呢？

欢迎你给我留言，和我一起讨论。

上一篇　077 | 推荐系统评测之三：无偏差估计

下一篇　079 | 现代推荐架构剖析之二：基于多层搜索架构的推荐系统

精选留言 (4)

写留言

永夜
2018-04-09
1

架构是时间，空间，算力等多种东西的折衷，分析问题将可离线计算和必须在线计算的数据拆解，主要还是看每种选择可能性的潜力和提升量。现在往往几十ms的时限下，空间换时间，加cache，分级别离线算好直接取都比较常用

展开

林彦
2018-04-05
1

如果用户数量和物品数量太大，线下计算无法满足每天全部更新推荐一次推荐。

1. 可以根据用户访问的频率，优先计算访问频率高的用户的线下更新。尽可能满足更新频

率略高于用户的历史访问频率(或预测访问频率)。另外新用户和新物品一般第一次更新尽可能及时。...

展开

帅帅

2018-10-20

@林彦的第3条很喜欢

不是根据每个用户做推荐，而是先将用户做聚类；把(userId、itemId、rank)变成(userGroupId、item、weightdRank)进行训练，得到每个聚群的推荐结果；
cache只需要存储每个聚群的推荐结果、每个用户的聚类映射即可，每次查询，先查询用...

展开

极客星星

2018-04-02

计算量过大时的解决方法 1采样降低数据量 2 模型几天才更新一次 3如果原来是单机计算改用分布式