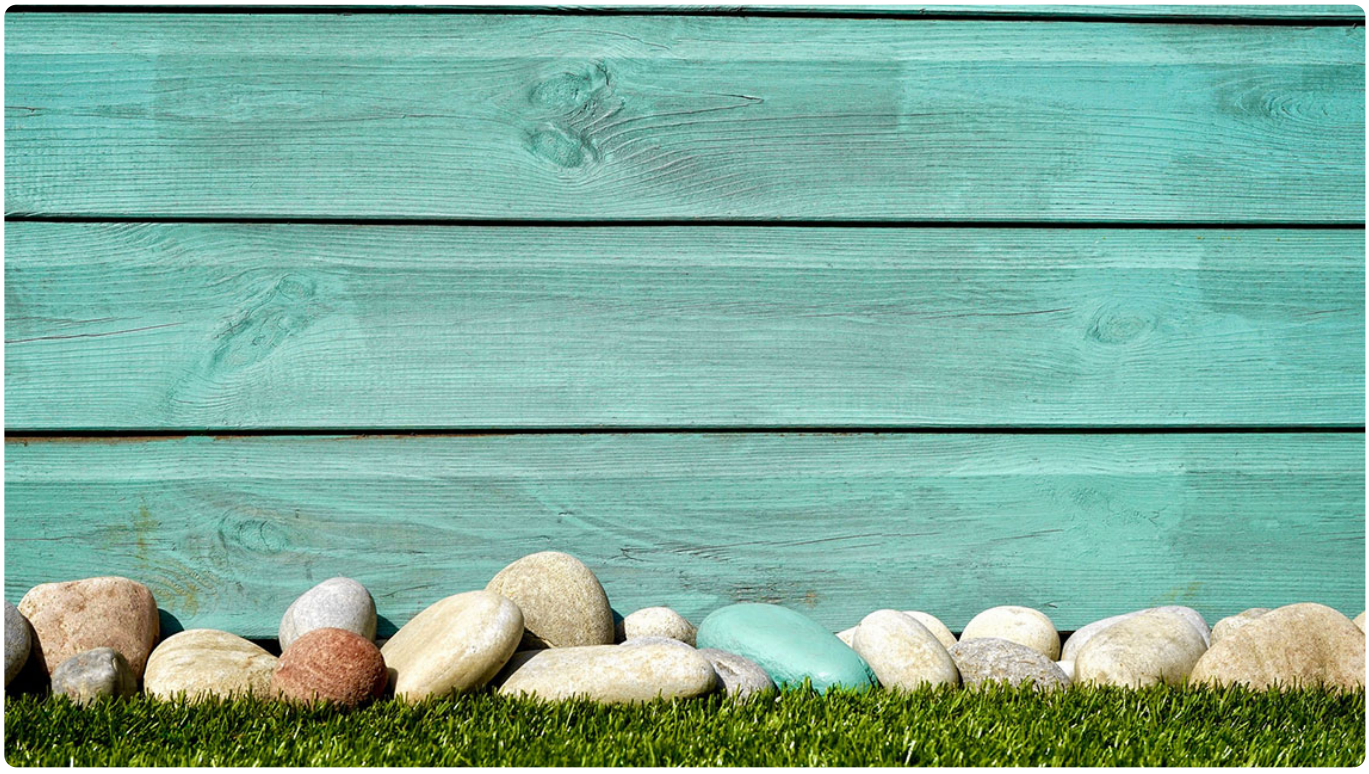
079 | 现代推荐架构剖析之二：基于多层搜索架构的推荐系统

2018-04-04 洪亮劼

AI技术内参 进入课程 >



讲述：初明明
时长 06:17 大小 2.88M

周一，我们讨论了基于线下离线计算的推荐架构，这也是最简单的一种推荐架构。我们了解了这种架构的优劣势，以及能够做的一些方案。

今天，我们来看另外一种也很常见的推荐系统架构，那就是**基于多层搜索架构的推荐系统**。

推荐架构需要解决的问题

周一我们详细讨论了推荐架构需要解决的问题，今天做一个简单的回顾。

推荐系统解决三个需求。

第一，推荐系统架构能够在一两百毫秒内给用户提供当前的推荐结果，即实时呈现推荐结果。

第二，推荐系统架构需要对用户和系统的交互结果做出响应。

第三，推荐系统架构需要考虑用户群体的覆盖率问题。

我们周一次讲到的基于离线计算的推荐架构，可以很好地解决第一个问题。解决思路就是先计算好所有的结果，然后存在某种存储空间里，当用户来到网站的时候，再直接显示事先已经计算好的结果。然而，这样的架构在第二和第三个需求面前，就显得有些捉襟见肘了。

基于多层搜索架构的推荐系统

我们前面在介绍搜索系统的时候，多次提到了**多层搜索架构**。一起来回顾一下这种架构。

首先，我们有一个索引，能够根据某些特性（比如关键字）来把所有的文档存储到里面，方便随时检索。

第一层或者叫第一轮打分，是发生在索引这个层面，我们通过一些简单的流程或者函数，往往是**线性函数**或者**布尔值函数**，来获取最相关的几百最多几千个文档。

紧接着，第二层或者叫第二轮打分，就是一个重排序的过程。这个时候，我们往往只需要针对几百个文档进行打分，所以可以使用相对比较复杂的函数，比如**基于决策树的模型**或者**深度模型**，以得到最终的结果。

有些时候，在第二轮打分之后，还有后面的轮数，主要是针对一些不同的商业规则，比如结果的多样性等等。

多层搜索架构可以支持搜索结果，自然地，对第一个需求，也就是在规定的时间内返回结果，有很好的支持。在搜索里面，用户输入查询关键词以后，大多数情况都希望能够快速得到结果。一般来说，我们把在所有文档里查找相关信息分解为两个步骤，先查找一个大概相关的集合，然后再在这个集合里进行重排序。特别是第一个步骤，往往是在索引上并行进行的，因此速度也相对比较快。

那么，多层搜索架构如何来解决第二和第三个需求呢？

我们先来看第二个需求，也就是说如何针对用户的反馈对结果进行更新。所谓进行更新，其实就是说，给用户的推荐结果，需要有一些不一样的地方。但是，如果我们仔细想一下这个需求，就会发现，第二个需求的核心是**需要对用户的反馈进行更新**，但也不能走向另外一个极端，那就是用户点击或者浏览了一两个物品后，整个推荐结果就全部发生了改变。因此，如果我们在这种**需要变化但又不是大变的假设**之下，多层搜索架构就能相对容易地解决这个问题。

例如，我们可以根据索引返回用户可能喜欢的一千个物品。假定用户的喜好不会在每一天内发生巨大变化。这个索引本身可以每天更新，但不需要更新得特别频繁。因为用户点击了一些物品，之后需要产生的更新变化，我们可以寄希望在重排序这个环节发生。也就是说，**我们在每一天中，从索引中提取出来的内容都可以是一模一样的，但是我们可以根据重排序的部分产生不一样的结果，这样也就满足了用户的需求。**

具体来说，在重排序的阶段，有两种方法可以根据用户的反馈进行更新。

一种方法，就是更新重排序阶段的模型。如果重排序阶段是一个决策树模型，那我们就对这个决策树进行重新训练。这里主要取决于重排序阶段是一个什么样的模型。如果这个模型需要所有用户的信息，那重新训练的计算量，无疑是非常大的，而且往往还无法在线完成。在这样的情况下，重新训练可能并不是最优的解决方案。

另外一种方法，就是更新重排序的模型的某些特性。如果重排序模型使用了一些特性，其中包含记录了用户的一些行为。那么，我们其实可以在不更改模型的情况下，通过更新特性的数值来达到更新结果的目的。比如，可能有这么一个特性，记录用户在某个物品上点击了多少次，那么我们单单刷新这个特性的数值就可以了。

对于第三个需求，也就是说如何针对新用户和新物品进行支持。可以说，**搜索架构对于新用户是天然支持的。**因为索引里面是物品，而并不是特定的用户信息，所以新老用户对于这个以索引为基础的架构来说是一样的。不太相同的自然是新老用户的特性值是不一样的，因此取决于重排序的模型，很有可能是针对老用户有比较强的效果，而针对新用户则可能会有一些捉襟见肘。

相对来说，**搜索架构的短板在于对新物品的支持。**因为整个索引机制是基于物品的，因此当我们已经建立了一个当前的索引后，新的物品不在索引里面，因而无法在提取阶段被取出来。一个比较直接的方法自然是重新建立索引，然而如果我们有上百万的物品，重建索引并不是一个简单容易的步骤。关于如何支持这样一个功能，我们留到下一次分享中探讨。

小结

今天我为你讲了基于多层搜索架构的推荐系统。

一起来回顾下要点：第一，我们回顾了推荐架构的需求；第二，我们介绍了什么是多层搜索架构，以及这个架构是如何利用到推荐场景的，同时还聊了聊这种架构的优缺点是什么。

最后，给你留一个思考题，我们谈到了用索引来帮助推荐系统的构建，那么在搜索里面索引可以根据关键字来建立，在推荐系统中，我们怎么构建索引呢？

欢迎你给我留言，和我一起讨论。

上一篇　078 | 现代推荐架构剖析之一：基于线下离线计算的推荐架构

下一篇　080 | 现代推荐架构剖析之三：复杂现代推荐架构漫谈

精选留言 (4)

写留言

永夜
2018-04-09

在我们实际应用过程中，越是近期行为尤其是用户刚刚触发的行为，指向的推荐效果最好。看了您的文章，我意识到还有一个行为置信度的问题。

damonhao
2018-04-06

推荐系统根据用户画像构建索引

展开

林彦
2018-04-06

除了搜索引擎用到的关键字外，可以使用用户或物品的属性和类别，如标签，主题，类簇，潜在语义做索引，还有人物，地理位置，书名，影视剧，历史事件和热点事件等实体也可以用来做索引。

展开

微微一笑
2018-04-04

对于构建索引，我尝试过将倒排信息存储在hbase，发现在召回阶段需要多次查询，效率不高；现在正在尝试将倒排信息存储在redis中。请问老师有什么建议？

展开