开篇词 | 从企业级项目开始，进阶推荐系统

2023-04-10 黄鸿波 来自北京

《手把手带你搭建推荐系统》

你好，我是黄鸿波。

先简单地介绍一下我自己，我目前就职于国内某大型游戏公司，主要负责 AI 团队的建设。同时我也是国内 40 多位谷歌开发者专家之一，《TensorFlow 进阶指南：基础、算法与应用》一书的作者。

从 2014 年开始，我就一直在从事算法和深度学习研发方面的工作，并带领团队为公司开发了推荐系统、智能问答、游戏强化学习对战机器人等产品，从 0 到 1 参与过很多推荐系统项目。

当今时代，任何产品都离不开推荐系统的加持，无论是信息流产品还是电商产品，基本上目前都是推荐为王的状态。因此，企业对于推荐系统的需求将会越来越大，对于推荐算法工程师的要求也会越来越高。

在推荐算法刚刚兴起的时候，基本上只要懂一些推荐系统算法的理论，就能够找到一份推荐算法相关的工作。但现在，只有对推荐系统整个的运转流程有了足够的了解，才能够获得企业的青睐。

在真实数据中成长

不会处理真实的数据、面对不同的问题不会变通，这基本是初学算法工程师的通病。造成这种局面最主要的一个原因，在于市面上很多入门课程，理论和算法虽然讲得特别精，但是算法与算法之间割裂非常严重。

并且由于这些课程用来练手的 demo 不是真实项目，数据集都是给造好的，基本上直接就能用（数据集给得一般比较精，没有太多冗余的数据），所以通常能够跑出非常好的结果。这样就会给人一种错觉：我算法学得挺熟，系统用 demo 数据集能够跑得很好，已经可以出师了。

但是真到了实际的项目里，“一学就会，一用就废”，这是比较打击我们自信心的。我们不妨一起看看在真实的项目中，你会遇到的三大难点。

难点一，外围知识多。

对于推荐系统来说，我们除了推荐算法本身的知识外，还要去学习和了解如何对前期的数据进行处理。当推荐的数据需要落地的时候，还需要学习内存型数据库以及多线程和多进程相关的知识。外围知识多，也就导致了想要学习好推荐系统，光掌握算法远远不够。

难点二，算法选型难。

一方面，新出的模型很多，各种论文效果也都非常好。另一方面，很多大公司在分享时使用的是比较保守的模型，激进和保守做法各有优势，算法选型需多多斟酌。一旦选错算法，会对后面的推荐效果有比较大的影响。

难点三，系统上线难。

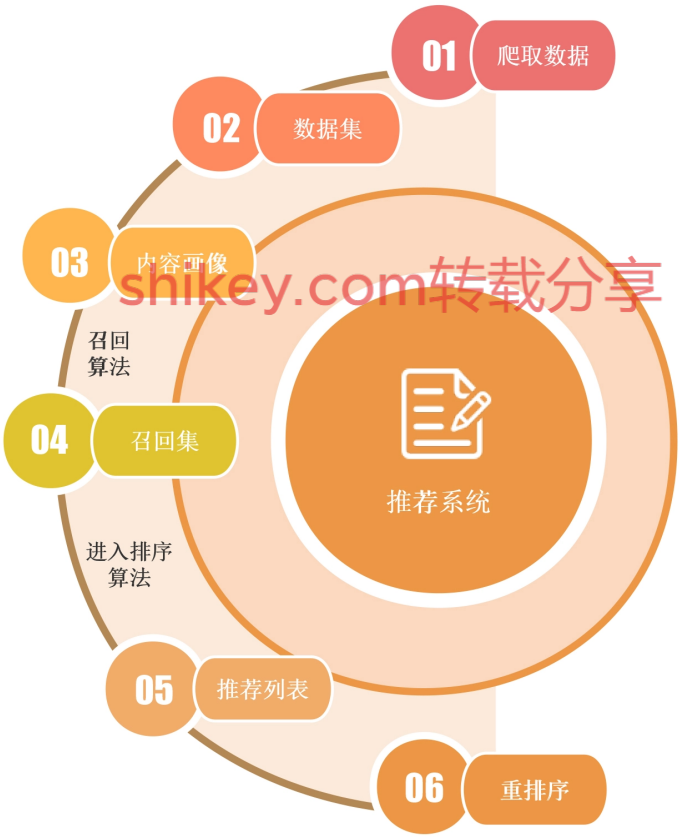
有的公司可能会有单独的工程化的同学，但是大部分都是基于传统项目的工程化，对于 AI 的模型上线以及相关的框架可能并不了解，导致企业中推荐系统上线会变得非常困难。

从上面的三个难点你能看出，推荐系统中有太多特征工程相关的内容，也有太多对工程和性能的要求。要学好推荐系统，不能只是简单地求快。我的这套课程的设计思路实际上很简单，用一句话总结：“**稳中求进，用真实数据搭建企业级系统**”。

算法 + 工程，平滑学习曲线

在这门课程中，我希望给你展现的不仅仅包括推荐算法，还包含了一整套推荐系统的工程化方案，也就是算法和工程双管齐下，让你在一个循序渐进的过程里，扎实走好每一步。因为在真实的生产实践当中，你不是一个人在战斗，你要和你的上下游一起去做一套完整的系统，要关注推荐系统算法对整个工程的影响。

在我们推荐系统的搭建过程中，一个特点是**环环相扣**。所谓环环相扣，就是指我们在前面所做的工作，一定会被后面所用到。比如，我们爬取的数据会被我们做成数据集，数据集会做成内容画像。而内容画像会用在召回算法中，并产生召回集。产生的召回集又会进入到排序算法，根据用户画像的内容，生成用户对应的推荐列表，然后再根据推荐系统的需要，对排序后的内容进行重排序。



极客时间

另外，既然是一个企业级的推荐系统，那么我希望这套系统应该是有一个界面的，所以，我会给你提供一个带有登录、注册、推荐列表、推荐内容详情和点赞、收藏等基础功能的 Web 界面，这个界面可以有助于我们收集用户行为，从而产生用户画像，使我们的推荐更加精准。

课程设计

对于一个完整的推荐系统来说，它应当包含的是从数据的获取、数据的处理、特征的采集、内容画像、用户画像，再到把这些画像和特征送入到算法层面，获取到我们想要的结果，以及对这些结果的存储和最终推送给用户的一整套流程。

从整体逻辑上来看，我将从最开始就带你获取真实的原始数据，然后把这些真实的原始数据处理成我们所需要的数据集和画像系统，最后再利用我们处理好的数据集和画像，搭建我们的推荐系统。

因此，我一共设计了七个大的章节，分别是：架构篇、数据篇、召回篇 - 基于规则的召回、服务搭建篇、召回篇 - 经典召回算法、排序篇和部署篇。

架构篇

在第一章，我将给你提供一个了解推荐系统概念和功能的宏观视角。你会了解到一个典型的推荐系统包含哪些重要部分，如召回、排序和个性化推荐等。架构篇会以 Netflix 系统为例，带你了解推荐系统的工作原理，为你展示推荐系统的运作流程和优化策略，从而让你对推荐系统的实践应用有一个整体认识。

数据篇

数据对于任何系统都是非常重要的，推荐系统也不例外。在第二章，我们将深入探讨推荐系统所依赖的数据处理流程。我们会先学习爬虫和数据库的原理和使用方法，然后开始实战，从新闻网站中爬取数据，将它们作为我们的原始数据集。接下来，我们还要使用 NLP、Python 等技术对数据进行简单的特征工程处理，形成我们的内容画像系统。

召回篇：基于规则的召回

接着，我们就进入到了推荐系统的召回部分，这是推荐系统的核心之一。我们将利用上一章得到的数据，进行一些基于规则的召回。基于规则的召回是一种简单而有效的召回方法，它基于时间、热度和关键词等有规律性的信息来为用户推荐内容。

在讲解什么是召回、为什么需要召回以及召回的种类的同时，我们会深入研究召回对于推荐系统的影响，以及如何选择最优的召回策略。

服务搭建篇

当我们有了一套可用的召回层数据后，就可以着手搭建一个推荐系统服务了。在本章，我们会将数据拼装后，绑定到界面进行内容推荐。我会给你一个简单的推荐系统 Web 界面，带领你在这个界面的基础上调用 Flask 提供的 webservice 接口，完成内容推荐。

召回篇：经典召回算法

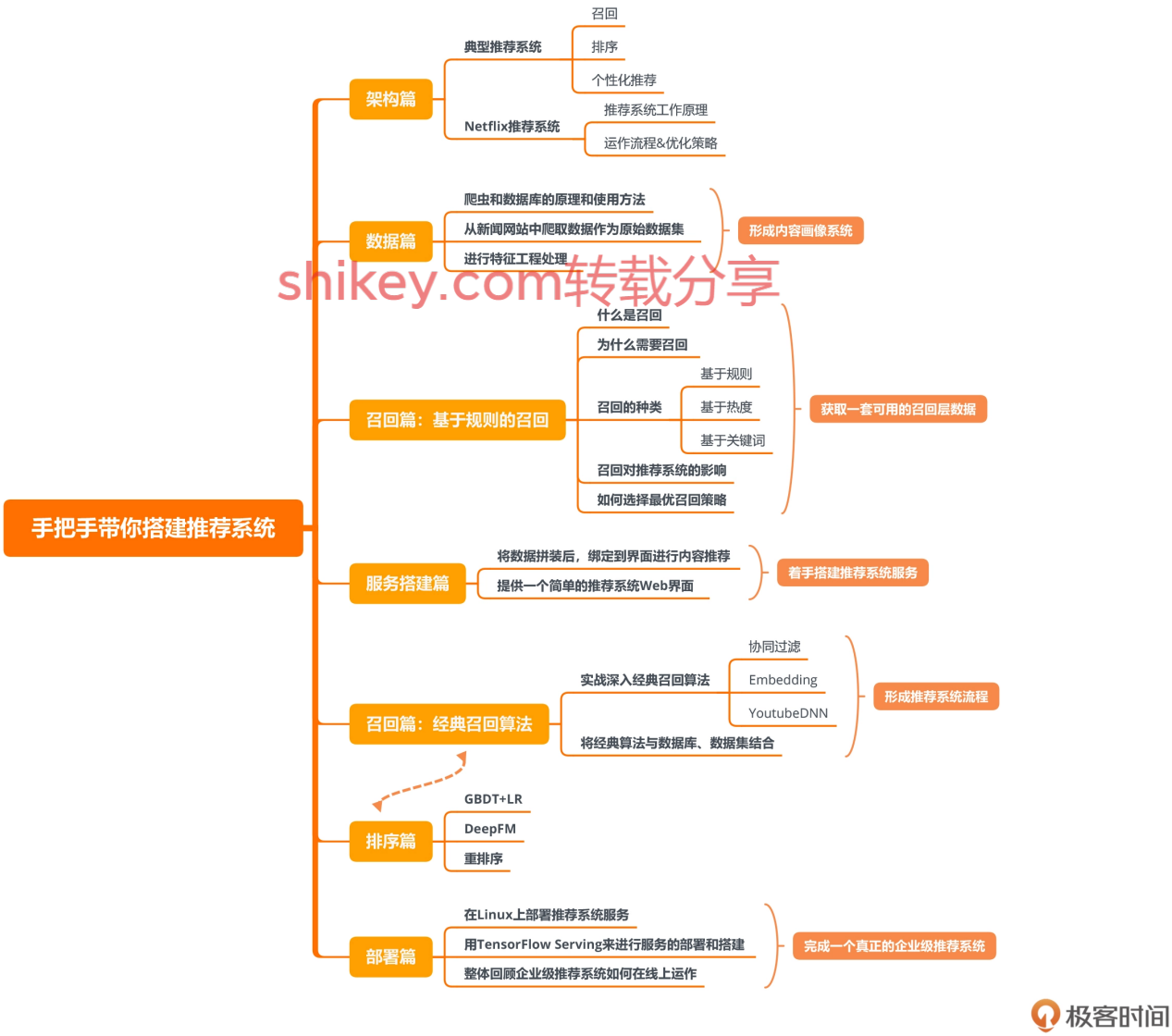
搭建好服务后我们更进一步，以实战的方式深入经典召回算法，形成一套简单的推荐系统流程。在这一章，我们将深入探讨包括协同过滤、基于 Embedding 的召回以及基于深度学习的召回等一系列经典召回算法，针对这些算法，我们会做不同的特征处理，并将它们与数据库、数据集结合起来。

排序篇

在推荐系统中，选出用户感兴趣的内容，并按照用户感兴趣的程度进行排序是非常重要的一环。在本章中，我们将讲解不少经典排序算法，包括 GBDT、LR、DeepFM、重排序等，充分利用现有数据，与上一章的推荐系统流程结合，用真实案例加深你对这些算法特点及适用场景的理解。

部署篇

到现在，我们已经拥有一个功能健全的推荐系统服务了。最后，我们尝试把整个系统部署和发布到 Linux 系统上，这也是我们的最终目标。我会带你使用 TensorFlow Serving 来进行服务的部署和搭建，完成一个真正的企业级推荐系统。在本章中，我们还会对推荐系统进行一个整体的回顾，从全局视角来观察企业级推荐系统是如何在线上运作的。



在带领团队做推荐系统的这几年里，我踩过不少隐藏的坑，比如说下面这几个。

> 在测试的时候效果非常好，但在上线后的效果就非常差。
>
> 在上线之前推理速度非常快，并且在整个测试的过程中也能够满足需求，但是上线一段时间之后，推理速度变得越来越慢。
>
> 在线上推荐时，推荐一些很久远的内容，或者已经被删掉的内容都被推了出来。

踩坑的代价就是你得一次又一次地加班，一直在制造 bug 和修复 bug 之间徘徊。如果我们想要少踩坑、少走弯路，别人的经历和教训是一个很好的学习样本，这也是我开这门课的初心。希望我的经验能够帮助到你，让这门课程成为你进入推荐系统的一块敲门砖，一起迎上时代的浪潮！

最后，欢迎在留言区分享你对于推荐系统的看法、对这门课程的期待。当下的技术趋势瞬息万变，未来将会属于那些有热情和创造力的人，下节课见！

精选留言 (5)

徐曙辉

2023-04-10 来自湖南

从课表上知道这门课后期待已久，从上个月盼到这个月，公司产品推荐系统逐渐承载不了需求，请教下老师如果小公司搭建推荐系统对硬件成本是不是要求比较高，没有专门的人力负责这块，人员成本和经济成本有限的前提下有没有比较好的解决方案，我想这也是小公司痛点。

作者回复: 同学，你好，其实对于小公司或者小团队来讲，搭建推荐系统的要求并不需要特别高，首先我们要评估好用量。

在我们的这套课程中，或者说真实企业的项目中，一共会分成下面几个部分：

1、推荐系统的模型项目：主要是推荐系统的算法运算、模型运算、推理等，这个可以单独用一个服务器；

2、前端的服务器，这个可以单独一个服务器，这个服务器要求不高，主要是web请求页面；

然后就是几个数据库，redis，MongoDB，以及MySQL（如果需要的话），这几个部分可以和前端共用一个服务器，也可以单独分开，这个就看预算了；

至于开发人员和成本，我觉得2~3个人的小团队，有2个月左右，一般来讲问题不大，如果前期没有很好的数据，可以尝试以冷启动或者我课程中的基于规则的推荐来启动这个项目。

共 3 条评论 >

3

卖代码的梦想家

2023-04-11 来自广东

之前推荐系统中用 Elasticsearch 作为召回组件，想问问老师这种做法和自己做的推荐模型区别在哪呢？

作者回复: Elasticsearch主要是基于关键词的相似度来进行推荐，这种推荐方式比较依赖于分词方法，分词的好坏决定了整体的好坏。

自己做推荐算法是利用了用户的特征和内容的特征，这里不仅包含了相似度，还包含了一些隐藏的语义和相关信息，相对来讲会更加准确一些。

 3

风轻扬

2023-04-10 来自北京

想啥来啥，最近正想系统了解一下推荐系统，极客就给安排上了，非常期待老师的更新。
另外，想请教一下老师，我查了一下flask，是python的一个框架，爬虫啥的也都是python，不咋懂python的研发，能学这门课吗？平时主要用java开发

作者回复: 我个人觉得是没问题的，我们的课程代码分成3个部分。

1、数据获取：这部分是讲爬虫，我们是用Python来写的，用的是Python中的scrapy框架；

2、推荐模型搭建：这部分是讲各个模型怎么写，以及对数据库的操作；

3、服务端的搭建：这部分是webservice。

在这三部分中，第一部分和第三部分，实际上用任何语言都可以开发，Java中也有很好的爬虫框架，也有比较好的服务端框架，比如springboot等；
那么第二部分模型篇，其实用Java也可以实现，只不过对于机器学习来说，Python的库更多一些。我们只要了解算法本质，其实语言只是一个工具而已，不用过于纠结语言。

共 2 条评论>

 2

- Forward

2023-05-14 来自广东

这门课为什么不讲讲大数据平台，我看到很多企业项目都用用到了大数据平台，如hadoop和spark等

作者回复: 同学你好，由于课程篇幅有限，以及很多时候大数据平台不是算法工程师关心的重点，所以在这里我们就没有去讲。

小嘟嘟

2023-04-11 来自上海

开篇很赞，非常具有吸引力，求更新

作者回复: 感谢支持。