

阿里云开发者社区

飞天大数据平台
APSARA BIG DATA PLATFORM

飞天AI平台
APSARA AI PLATFORM

个性化推荐系统 开发指南

机器学习PAI从入门到精通

作者：傲海



召回、排序推荐算法+线上服务编排超详细解析
阿里云技术专家手把手教你搭建智能推荐引擎



阿里云开发者电子书系列



阿里云开发者“藏经阁”
海量免费电子书下载

I 目录

推荐系统简介	4
推荐系统召回算法	10
推荐系统排序算法	14
推荐系统线上服务编排	18
基于 PAI 10 分钟搭建一个简单推荐系统	21

推荐系统简介

一、什么是推荐系统

(一) 常见的推荐业务场景

首先给大家分享一下什么是推荐系统，为什么要有推荐系统。伴随着互联网应用的发展，人们可以涉猎到更多的资讯。比如说进入到一个淘宝的平台，有非常多的商品，如何将适合用户的商品去触达他，是淘宝需要解决的一个问题。本质上，推荐系统解决的是一个信息比对的问题。怎么样**基于用户的信息和商品的信息去做一个更好的匹配**，这是推荐系统要解决的问题。大家平时在用各种手机 APP 的时候，其实已经对很多的推荐场景或多或少的有些了解。常见的推荐业务场景有两个。一个是基于搜索 Query 的推荐，还有一个是基于用户和商品属性的 Feed 流的推荐。

大家看左边这个图，什么叫 Query 推荐，比如说我搜索一个口罩，它下面展示的肯定都是跟口罩相关的东西。可能有 11 万多个口罩相关的商品，哪些排在前面，哪些排到后面，这里需要有推荐系统。它需要根据用户的属性，比如说他喜欢的颜色、价格偏好等，来进行排序。如果他喜欢买贵的奢侈品，我肯定把一些贵的、性能好的口罩放在前面。如果他是一个价格敏感的用户，我可能要把价格稍微便宜、性价比高的口罩放在前面。综上，就是说 Query 推荐要基于用户的购买偏好，还有商品的属性去做一个匹配。

再看右图，Feed 流推荐越来越成为很多 APP 跟用户交互的主流。打开虎扑、今日头条这样的 APP，你会发现首页的新闻都是根据你日常的偏好去推荐的。比如说你喜欢篮球相关的新闻，它可能更多的把体育相关的内容推荐给你。基于用户和商品的 Feed 流推荐，我们采用机器学习推荐模型，它既要学习用户，也要学习商品的属性。我们今天介绍的推荐系统架构，在用户属性和商品属性的匹配过程中，底层的系统实现，更多的是偏向于基于用户和商品属性的 Feed 流推荐这样的一个场景。

常见的推荐业务场景



基于搜索Query的推荐

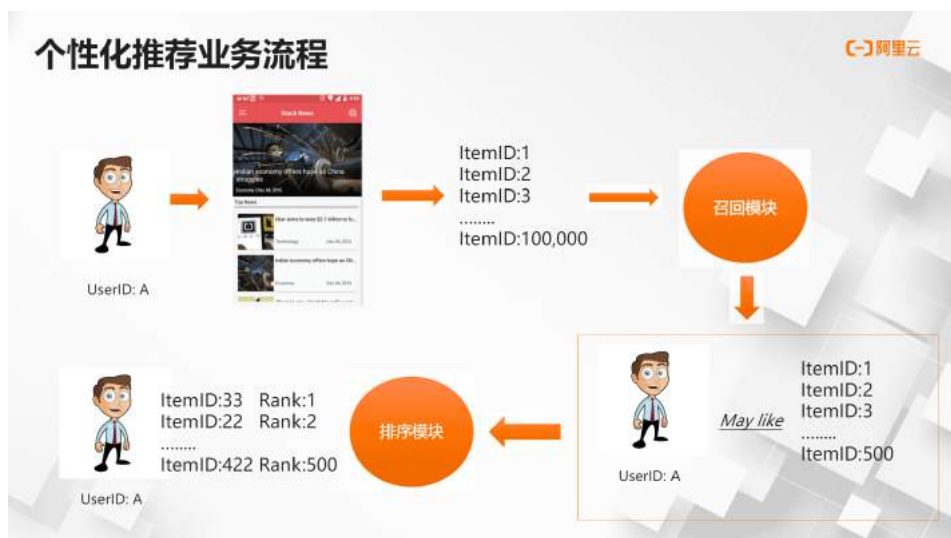


基于用户和商品属性的Feed流的推荐

(二) 个性化推荐业务流程

首先我把整个推荐业务做了一个简图，如下图所示。假设我们有一个新闻平台，用户 A 进来，它有一个 ID。这个平台有成千上万的新闻，我们把它叫 item。每个 item 有一个 ID，比如说 1、2、3 这样排列下去。我们现在要把 10 万个 item 中筛选出用户 A 喜欢的新闻。我们看一下要做这件事，在底层的业务架构要有哪些模块。在一个经典的排序推荐召回系统里，它会有两个模块。第一个叫召回，第二个叫排序。召回模块把这 10 万个新闻做一个初筛，选出 A 可能喜欢的，比如说 500 个 item 先放到这里，我们只知道 A 大致会喜欢这 500 个 item，并不知道这 500 个 item 哪个是 A 最喜欢的，哪个是他第二喜欢的。接下来是排序模块，对这 500 个 item 做一个排序，按照喜爱的顺序去制定最终投放给 A 的 item 的列表。所以在整个的推荐业务中，召回模块更多的是一个初筛，确定一个大体的轮廓和范围。这样的话可以加速排序模块对于每个商品的属性排序，用户得到推荐反馈的效率会更高。一个专业的推荐系统需要在几十毫秒内，即用户一进来就可以给他推荐反馈。对 Feed 流的内容，用户在网上刷一下屏幕，可能有几十毫秒，然后我们马上要把新推荐的 item 又展示出来。这是整个推荐业务的逻辑。

推荐系统可以理解为推荐算法和系统工程的总和，即**推荐系统 = 推荐算法 + 系统工程**。关于推荐系统，很多的书和网上的资料更多的是聚焦到这个算法怎么做，包括很多 paper 都说的是最新的推荐算法。但是，当你真正动手去搭建这套业务系统，特别是在云上去做的时候，你发现其实是一个系统化的工程。即使你知道推荐业务需要用哪些算法，你依然会面临很多问题。比如说性能的问题、数据存储的问题，等等。所以我们这一次分享的重点，把系统工程的问题也给大家阐述出来，既讲算法也讲系统，两者结合就是一个完整的推荐系统。



二、企业级推荐系统架构

(一) 企业级推荐系统要求

接下来我们看一下整个的企业级推荐系统架构。设计这样的一个架构，有四个基本要求。第一个要求，目标客户有百万级 MAU (monthly active user) 的一个推荐业务需求的应用，每一次参与模型训练的时候，整个的**训练样本量可能是上亿级别**的。需要基于整个平台过去一个月甚至半年的数据做一个整体的建模，因为在机器学习领域数据量越大，模型越精准。数据可以拆分为三种：用户的行为数据、商品的行

为数据、用户商品之间的交互数据。第二个要求，它要有**算法插件化部署**的能力。大家知道机器学习这个领域，包括推荐领域发展特别快，每年都会衍生出一些新的算法。这些算法在整个系统中能不能做到灵活地插拔。比如说今天我用算法 A，明天算法 B。我可不可以方便的把算法 A 直接踢掉。这就是你的系统的鲁棒性，包括对于算法组件化支持的能力。我们现在推出的阿里云机器学习平台 PAI 有这样的一个能力。第三个要求，就是服务的性能问题。你是否可以做到**每次请求毫秒级反馈**。第四个要求，就是**支持资源的弹性拓展**。比如说对于一些 APP，可能下班的时候大家在地铁上刷得比较多，凌晨大家刷得很少，你的推荐模型底层的使用资源，在用户量大的时候需要更多资源，在凌晨需要更少的资源。为了平衡成本，你能否实现底层资源的灵活拓展性。这个可能适用云上的服务，它的一个优势是资源的弹性。如果搭建一个企业级的推荐系统，一定要满足上述四个基本的要求。

企业级推荐系统要求



目标：百万+级 MAU 应用的推荐业务



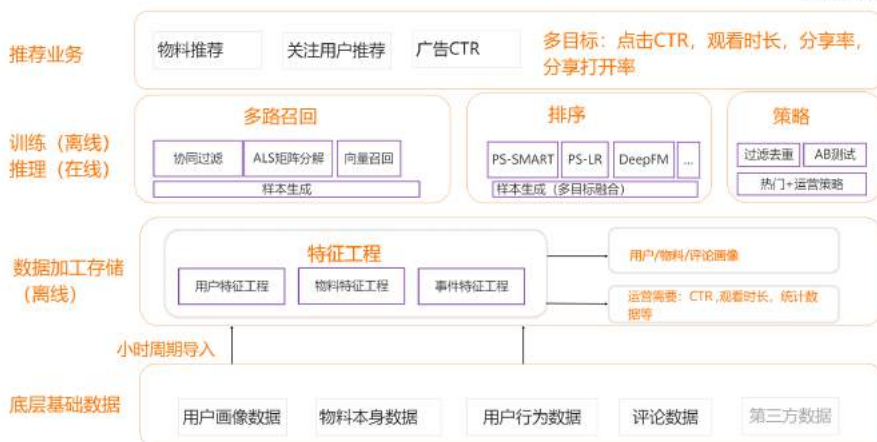
(二) 推荐整体架构

接下来我们重点讲一下推荐的整体架构。如下图所示，最下面就是基础数据层。我们可以看到有用户的画像数据，有物料本身的数据，行为数据，评论数据。用户画像数据可能是用户的身高体重，过去买过的东西，购买偏好，他的学历等等。物料数据就是说物品的价格、颜色、产地。如果是视频的话，视频的内容、标签等等都属于物料本身的数据。行为数据是指用户和物料之间的交互。比如说用户看了一个视频，

他点赞，收藏，投币。这些都是用户的行为数据。还有评论数据，第三方数据等等，可能不一定每个平台每个产品都会有。但是基本上这三个数据，user 的数据、item 的数据、还有 behavior 的数据是一定要有的。当我们有了这三份数据之后，就会进入到数据加工存储层。在这一层我们会做一些数据加工，比如说把用户的特征加工出来，把物料的特征加工出来，把这个事件的特征加工出来。再往上一层就是基于这些特征去建模。我们刚才介绍了整个的推荐流程包含召回和排序这两个重要的模块。召回模块中，可以有多个算法并行去做。

召回完之后你需要排序，也有很多算法，究竟选哪一种算法，后续第三节课再说。接下来，你要有一个新的策略，还不能把推荐结果直接拿到线上，要有一些过滤去重、AB 测试、运营策略。比如说我昨天刚推荐给你一个小米手机，然后你就买了。我今天再推荐小米手机肯定是不合适的。最上层就是推荐业务，可以推荐一个广告，可以推荐商品，也可以推荐用户。比如说在社交应用中，可以把用户推荐给用户，让他们互相关注。有了这一整套推荐架构，怎么样让它去符合企业级推荐系统的四个基本要求，需要应用到一些云产品。最常见的做法是，基于云服务、云生态去搭建这些模块。

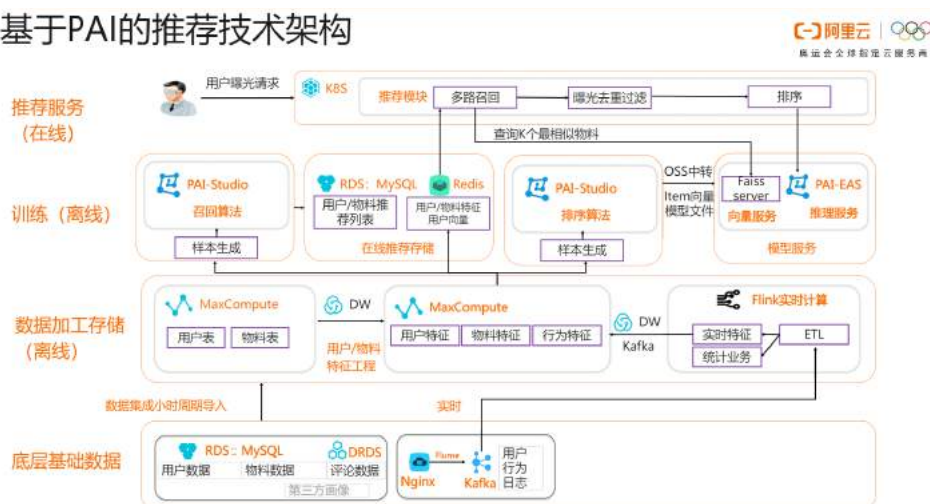
推荐整体架构



(三) 基于 PAI 的推荐技术架构

在阿里云这边的整个的实现方案是这样的。基于 PAI 的推荐平台，在基础数据层，我们可以提供给你网站的一些离线数据，可以存到 RDS::MySQL 这样的一个数据库里。线上的一些行为数据，比如说你实时有一些点击，有些关注，你可能想做一些实时的处理。你可以做 Kafka。然后到上一层，数据加工层你可以用 Flink 做一个加工，产生一些实时的行为数据。生成了样本之后，可以发到模型训练层。在模型训练层就会用到 PAI 的一些算法。最上面应用的时候，为了保证整个服务的弹性，我们推荐通过云原生的方案去做，保证资源的弹性。最终在服务编排阶段，你可以先调一下召回，拿到召回的结果，然后去重，拿到最终的一个给排序的样本，然后排序的结果就会反馈给用户。这是我们整体的技术架构。

基于PAI的推荐技术架构



推荐系统召回算法

一、召回模块在推荐系统中的位置

在如何构建企业级推荐系统系列课程的第一课中，我们介绍了整个推荐系统的架构应该是什么样的，大体有哪些模块，在云上的产品的每个模块需要用什么样的一个应用。在这一节课，我们会重点针对推荐系统中的召回算法做一个具体的说明，以及这个架构是怎么搭建的。先来回顾一下召回模块在整个推荐系统中的一个位置。召回模块主要做的一件事就是初筛。用户 A 进到平台中，从潜在的超大规模的 item 中筛选 A 可能喜欢的较小比例的 item。比如说平台上有 10 万个 item，用户进来之后筛选出 500 个他可能会喜欢的。接下来通过排序模块根据他的喜好程度对 item 进行排序。



二、推荐场景 - 召回算法介绍

今天我们就介绍一下在这个召回模块究竟会用到哪些算法。我把目前比较流行的 4 个算法放到这里。右侧的是协同过滤算法，左侧的可以叫做向量召回相关的算法。协同过滤算法类似于基于统计的算法，它会找出兴趣相同的一些人，或者说一些可以被同时购买的商品。比如说啤酒和尿布的故事，我们发现在超市里啤酒和尿布经常被

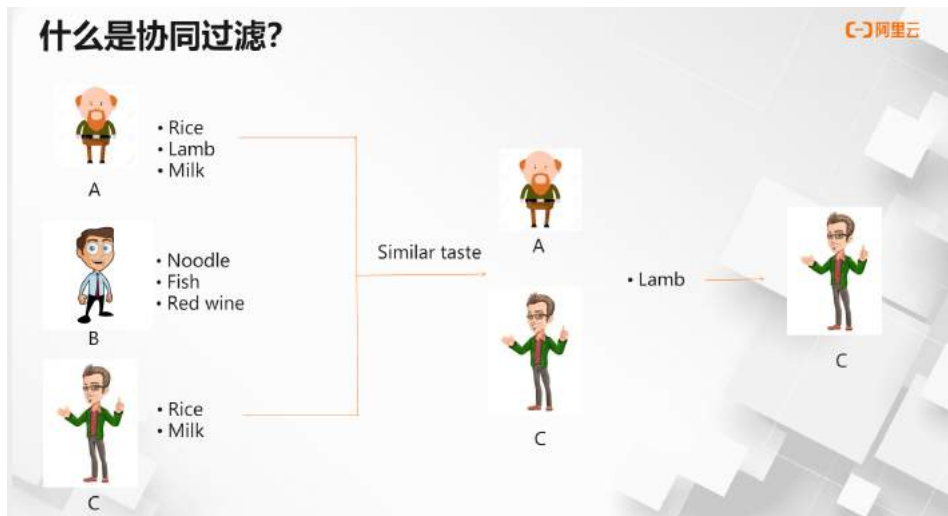
一起购买，这是基于大量的数据统计得到的一个结果。向量召回更偏机器学习的一些深层的模型。比如说 ALS 是一个经典的矩阵分解方法，可以基于行为数据表，产出 user embedding 表和 item embedding 表。这是向量召回的一个基本方法。其实 FM 也是类似的一个逻辑，通过内积方式增强特征表现力。这里面我要单独介绍一下 GraphSage 算法。它其实是一种图神经网络的召回算法，这种算法在整个的互联网领域应用的还不是特别多。但是在一些比较大的互联网公司，比如说淘宝的推荐场景会应用得比较多。GraphSage 是基于深度学习框架构建的图算法。可以基于用户和商品特征及行为产出 user embedding 和 item embedding。GraphSage 算法也经常被用于电商的召回场景。



三、什么是协同过滤

协同过滤可能是比较好理解的一个算法，我这里会用一个篇幅给大家介绍一下。比如说我们有用户 A、B、C，他们的偏好如下图所示。你会发现 A 和 C 有一些相似的口味。A 和 C 都喜欢 Rice、Milk。除此之外，A 还喜欢 Lamb。就是说 A 和 C 有相似的口味，并且 A 还喜欢某个 C 没有尝试过的东西。我们就假设 C 也喜欢它，可以把它作为 C 的一个召回结果。这就是一个基于数据统计的标准的协同过滤的方式。

这张图可以帮大家很好地去理解协同过滤算法的原理。



四、向量召回架构说明

接下来再介绍一下刚才的三种向量召回的算法。它的一个输入的数据大体是这样的，你有一些用户的 ID，然后有一些物品的 ID，然后有一些行为的数据。如下图所示，这是一个用户行为的数据表，进来之后你通过向量召回算法，最终拿到的是两个向量表。这些向量表在实际的生产中是一个 K、V 形式的。就是一个用户 ID 对应一个向量，可以在 Redis 里去暂存一下。实际的使用过程中需要把这些东西全存到 Faiss 的服务器里。Faiss 是 Facebook 的 AI 团队开源的专门针对向量检索的一个引擎。它的一个好处就是非常快，可以提供很多的向量检索的模式，甚至可以在一毫秒之内返回一个几百万级别的向量检索，它的性能非常好，常用到推荐召回的领域里。具体召回的时候，比如说我们想给一个用户推荐一个物品，我们拿着用户的 ID 和它的向量进入到 Faiss 引擎里去查哪些物料向量和用户向量的欧式距离最近。比如说，把 Top10 个拿出来，作为这名用户的召回结果。这是整个的方案，会用到 Redis，也会用到 Faiss 这样的一个引擎。这就是整个的向量召回的架构。

向量召回架构说明

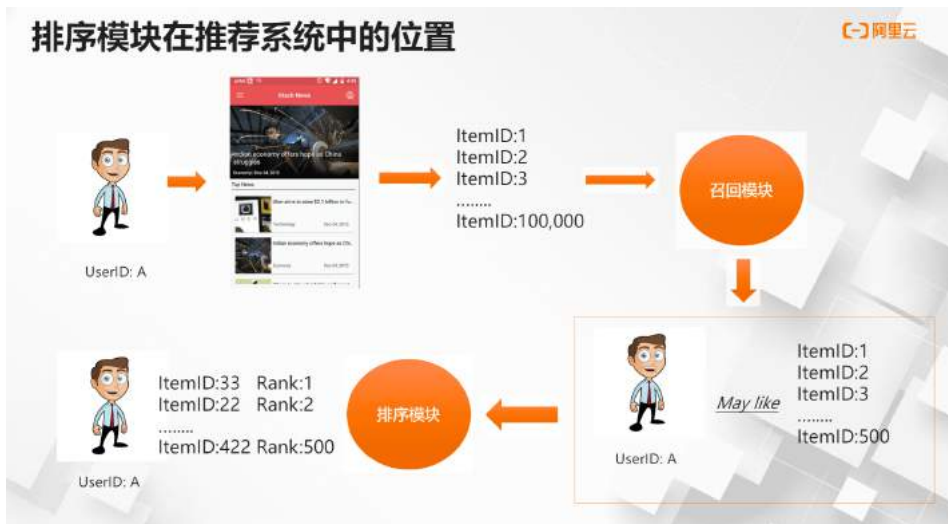


推荐系统排序算法

一、排序模块在推荐系统中的位置

在如何构建企业级推荐系统系列课程的第一课中，我们介绍了如何去设计整个企业级的推荐系统架构。在第二课中，我们介绍了怎么去理解召回算法以及召回算法的一些意义。在本节课中，我们分享一下推荐系统的排序算法，以及它的一些架构特点。首先，我们再回顾一下这张图，确定一下排序算法在推荐系统中的一个位置。一个用户进入到平台，会发现有很多商品，得把用户可能喜欢的商品先挑出来。我们通过召回模块，进行一个初筛，缩小排序模型的一个受众的范围。

召回模块从潜在的超大规模的 item 中筛选用户可能喜欢的较小比例的 item。比如说平台上有 10 万个 item，用户进来之后筛选出 500 个他可能会喜欢的。接下来通过排序模块根据他的喜好程度对 item 进行排序。从这 500 个 item 里怎么挑出用户最喜欢的东西，怎么挑出用户第二喜欢的东西，这时候需要一个排序的算法模型。它会针对这 500 个商品和用户的一些特点，即用户特征进行一个排序。



二、排序算法介绍

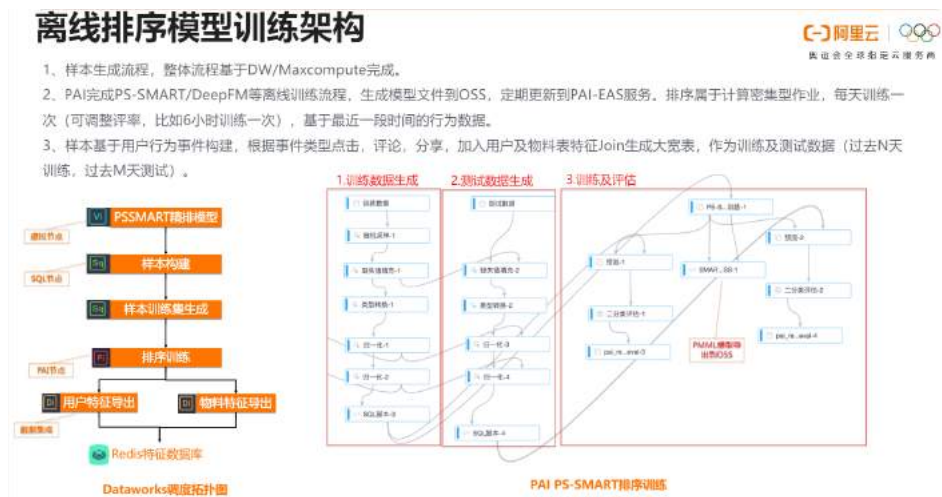
我们今天介绍一下排序算法究竟有哪些种类，排序的模型该怎么训练，它的系统架构是怎么样的。排序模型非常多，随着深度学习的发展，整个的排序算法已经从浅层的算法逐渐往深度学习迁移。目前比较经典的排序算法还是我列的这4种，如下图所示。首先，逻辑回归是应用非常广泛的一种算法。它是目前业内最经典的线性二分类算法，特点是容易上手，对于计算力要求低，模型可解释性好。第二，FM算法近一两年来，在很多客户的场景中都得到大规模的应用，效果也不错。它是通过内积的方式增强特征的表现力。第三，在LR基础上通过GBDT和特征编码增强数据特征的可解释性。第四，DeepFM算法也是现在被应用得非常多的一个深度学习的算法，它把深度学习和经典的机器学习算法做一个结合。如果大家刚开始尝试搭建一套推荐系统，建议先从简单的算法去尝试，后续再逐渐的往更复杂的模型迁移。当然，这些算法都已经内置到了PAI平台，已经可以插件化的应用。



三、离线排序模型训练架构

接下来讲一下排序模型该怎么训练产生，训练过程分为离线训练和在线训练，首先讲离线排序。

离线训练用的是 T-1 的数据，也就是说，今天业务上用的模型是通过今天以前的数据训练出来的。好处是可以把历史的大量数据几乎集合到一个数据仓库里，然后基于整体的全样本的特征和数据进行离线的训练，并且再做一个离线的验证。这样的话，生成的模型已经离线验证过了是没问题的。第二天直接把它推到线上去用，可以保证一定安全性和效果。这是一个离线的训练模型的架构。目前几乎所有的客户都采用离线排序模型的训练方案。如果你想做实时化的训练，会有比较大的架构挑战。



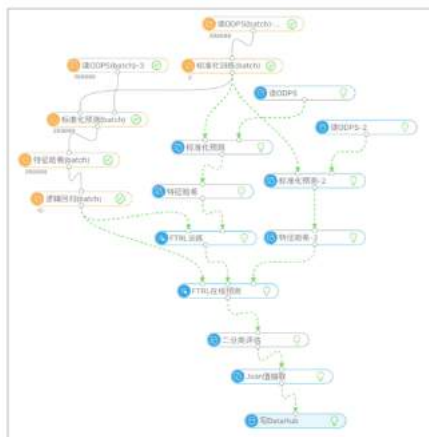
四、在线排序模型训练架构

我们今天重点放在在线模型训练这块。相比于离线训练，在线训练提供什么好处。离线训练毕竟使用的是 T-1 或者 T-2 的数据去做的，没有对线上实时产生的行为数据进行利用，对于数据的时效性利用相对较差。比如说，有这样的一个场景，今天我的整个平台只对 14 岁以下的少女做某个运营活动，而平台上充斥了大量的年龄段的客户，整个平台的交互行为都变了，这个时候你的模型还是 T-1 去做的，将无法对线上业务做一个及时的感知。这个时候，通过在线训练就能比较好的解决这个问题。在线模型训练的流程如下图所示。在线模型训练意味着我可以用实时上传的数据，然后用我们的实时机器学习模型训练框架去做训练。在线训练虽然数据是实时进来的，但你的模型并不是从 0 开始的。而是说我从离线先训练好这个模型，我站

在离线模型的巨人的肩膀上，再往上去优化。这就是在线排序模型训练的一个框架。它有 3 个特点。第一，它是基于 Flink 框架实现的流式模型训练能力。第二，它是基于实时生成的模型去实时评估模型效果。第三，它具备线上模型回滚和版本管理的能力。你要达到这三个标准，才可以构建一个完整的在线模型训练的架构。

在线排序模型训练架构

阿里云



- 1、基于Flink框架实现流式模型训练能力
- 2、基于事实生成的模型去实时评估模型效果
- 3、具备线上模型回滚和版本管控功能



推荐系统线上服务编排

一、在线推理服务 – 架构说明

在如何构建企业级推荐系统系列课程中，前两节课我们分别介绍了召回算法及架构、排序算法及架构。本节课我们会介绍一下如何对排序和召回算法的结果做一个服务编排，以及最终我们的模型跟现场业务是如何对接的。首先，我们介绍一下整个的框架，用户的业务场景，特别是互联网的推荐业务，它的高峰基本上会集中在中午和晚上。这个时候如何减少咱们的资源消耗。比如说，你的高峰需要消耗 10 台机器，低谷可能用 1 台机器。如果你没有弹性的机制，在线下买，一定要买 10 台机器才能满足服务高峰的一个需求。但是，如果你使用了云上的服务，就可以把这一套东西做到弹性。相比于你一直拥有 10 台机器，会节约很多成本，这是一个推荐业务需要考虑的场景。所以在服务编排阶段通常会采用云上的这套架构去做，来应对它的业务的一个潮汐效应。

另外一个要解决的问题就是，召回和排序这样的流程究竟该怎么创建起来。我们的方案是，基于高扩展弹性业务场景，采用阿里云 ACK 构建整体推理架构。调用流程分为 3 步。第一步，多路召回：物品协同过滤，语义召回，热门及运营策略召回取回上千条候选集。第二步，曝光去重：基于该用户阅读历史，去掉已经曝光内容，去掉基于运营策略不能推荐的内容。第三步，排序：推理模块调用排序过程时根据用户 ID 及物料 ID，获取用户特征及物料特征后，分批调用 PAI-EAS 服务返回排序结果。右面这张图其实讲的是排序 PAI-EAS 在线推理服务的一个在线监控的能力。就是你要观察排序模型的一个水位，进行动态的扩容、缩容，防止你的 RT 过高，或者 QPS 达不到业务的需求。这是整个的在线服务编排的框架。

在线推理服务-架构说明



- 1、业务场景：客户业务潮汐效应很明显，业务高峰基本集中在中午和晚上。
- 2、方案：基于高扩展弹性业务场景，采用阿里云ACK构建整体推理架构。
- 2、调用流程：
 - 2.1、多路召回：物品协同过滤，语义召回，热门及运营策略召回取回上千条候选集。
 - 2.2、曝光去重：基于该用户阅读历史，去掉已经曝光内容，去掉基于运营策略不能推荐的内容（敏感内容）。
 - 2.3、排序：推理模块调用排序过程时根据userId及物料id，获取用户特征及物料特征（Redis）后，分批调用PAI-EAS服务返回排序结果。



二、线上多目标问题

最终模型上线的时候，我们还会面临一个问题。我们设计整个架构，有的时候并不是单一目标，而是多目标的。就拿文章或视频推荐为例，很多的客户都希望说，不光是我推荐的视频被用户点击了，用户还需要多看一会儿，这样的话就会规避一些标题党。比如说有些视频把标题起得很吸引人，但其实内容很浮躁。用户点进去看一眼觉得没意思就走了，这样的话他整个的使用体验也不好。所以它可能是多目标的，既要他点击，也要他多看一会。针对这样的一个多目标情况，我们该怎么设计整套的方案，怎么去编排整套的推荐召回应用逻辑，有两种方案。一种是说多模型解决多目标问题。假设就是点击和时长这两个目标，你可以有一套推荐召回模块专门针对点击。

另一块专门针对使用时长去做训练。这两个结果你把它融合一下，得到最终的推荐结果。但代价就会比较大，你要同时维护两个系统，而且二者的比例也不好去量化。方案二是合并多目标成单模型，是目前采用得比较多的一个方案，也是效果相对来讲会比较好的一个方案。你把目标一和目标二这两个目标先融合成一个目标。比如说你把是否点击和观看时长按照一个比例去压缩下，把它都放到 0~1 之间。不点击就 0，点击就是 1。然后你把观看时长去做一个归一化，把整个时间都缩小到 0~1 的

区间去。这样，你整个的区间就变成了 0~2，变成一个单目标的数值。这样的话你就可以针对这一个目标去训练你的召回、排序模型，从而拿到最终的结果。这样做的好处是你只需要维护一套推荐业务的建模流程，会比较方便维护，最后的效果也通常是方案二好一些。

线上多目标问题



方案一：多模型解决多目标



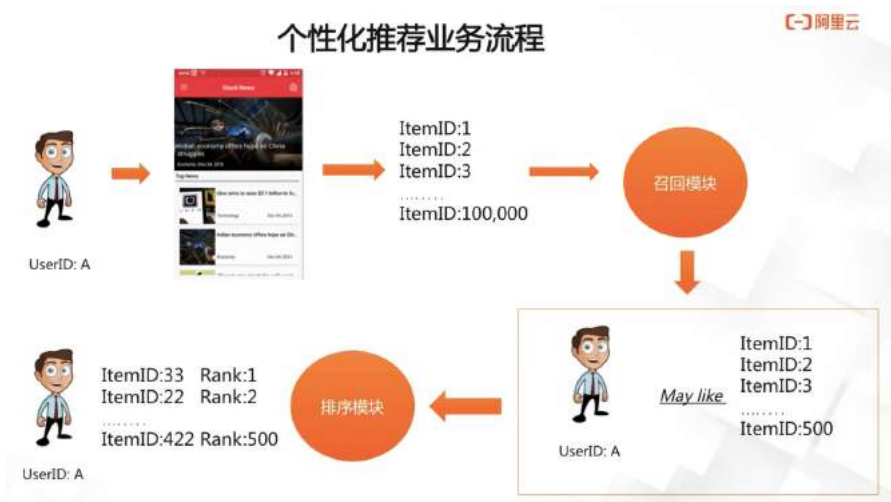
方案二：合并多目标成单模型



基于 PAI 10 分钟搭建一个简单推荐系统

一、个性化推荐业务流程

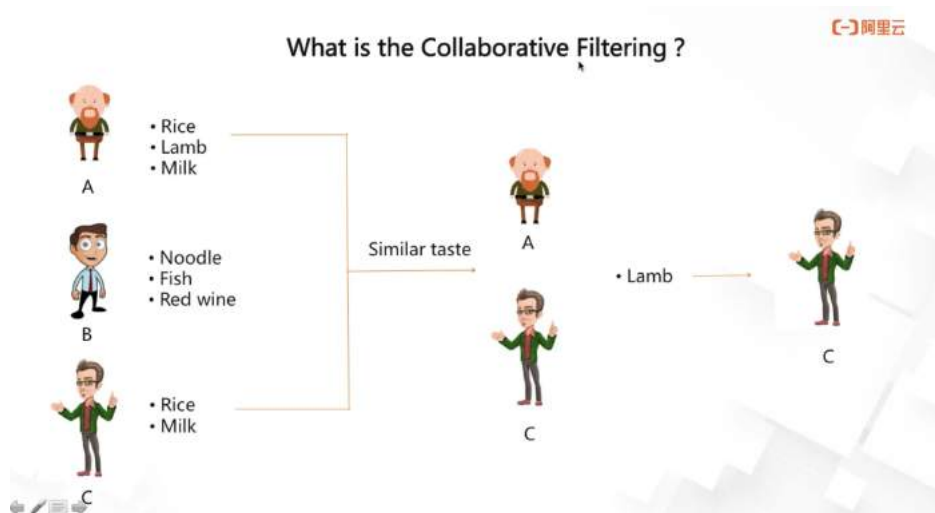
今天会为大家介绍一下如何基于 PAI 去搭建一套非常简单的推荐系统。这套推荐系统有两个特点。第一是搭建非常方便，因为我们做了很多工具。第二，这套系统是可扩展的。大家知道现在的 APP 都有很多 feed 流推荐，包括一些广告推荐、内容推荐，本质上就是一个推荐系统。推荐系统可以分为两个主要的模块。第一个是召回模块，它做的事情就是初筛。比如说，把 10 万个候选新闻缩小范围到只剩 500 个。第二个是排序模块，按用户的偏好程度对 500 个新闻排序，就可以生成最终的推荐顺序。在一个非常简单的推荐系统中，只有召回模块也是可以做推荐的。假设我把这个范围缩到足够小，比如说 10 个新闻，不需要排序，直接把这 10 个都推给用户就好。所以今天我们这个视频的内容就是教大家搭建一个只有召回模块的简单推荐系统。



二、协同过滤

召回和排序都有很多算法。比如说召回领域有很多矩阵分解算法、协同过滤算法等等。最经典的一个就是我们这里展示的 collaborative filtering，即协同过滤算法。

它可能是比较好理解的一个算法，我这里会用一个篇幅给大家介绍一下。比如说我们有用户 A、B、C，他们的偏好如下图所示。你会发现 A 和 C 有一些相似的口味。A 和 C 都喜欢 Rice、Milk。除此之外，A 还喜欢 Lamb。就是说 A 和 C 有相似的口味，并且 A 还喜欢某个 C 没有尝试过的东西。我们就假设 C 也喜欢它，可以把它作为 C 的一个召回结果。这就是一个基于数据统计的标准的协同过滤的方式。这张图可以帮大家很好地去理解协同过滤算法的原理。就是说，先发现相似的物品或者相似的人，然后根据相似的人和相似的物品之间的关联性去找出它的一个推荐逻辑。我们今天的这一套推荐系统就是以协同过滤算法为基础去做。



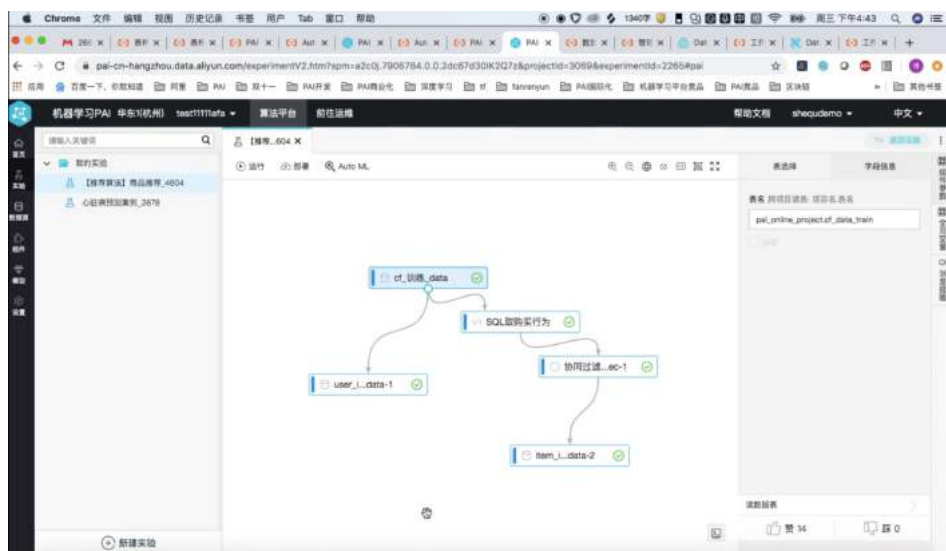
三、推荐方案架构

完整的推荐架构如下图所示，包含如下部分：Dataworks，PAI-Studio，TableStore，PAI-AutoLearning，PAI-EAS。

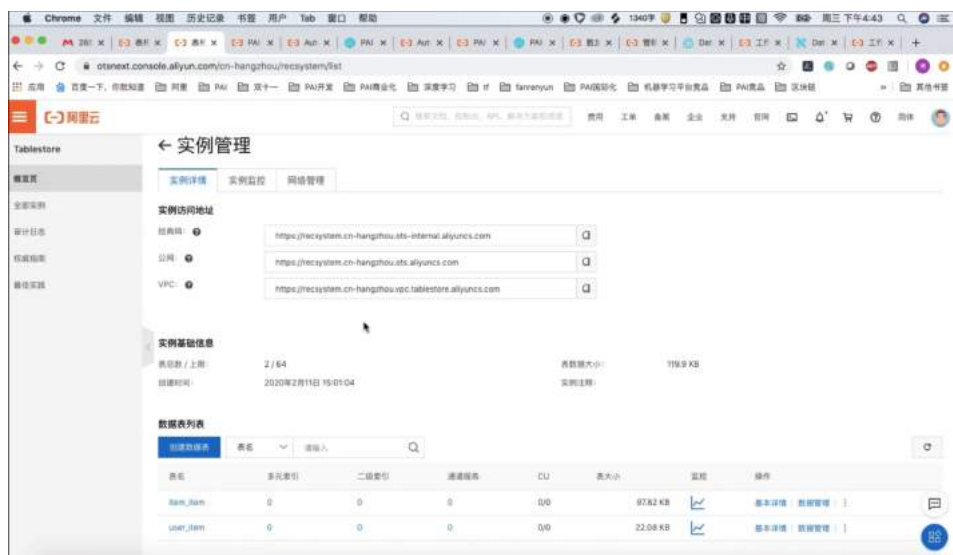


四、实际操作

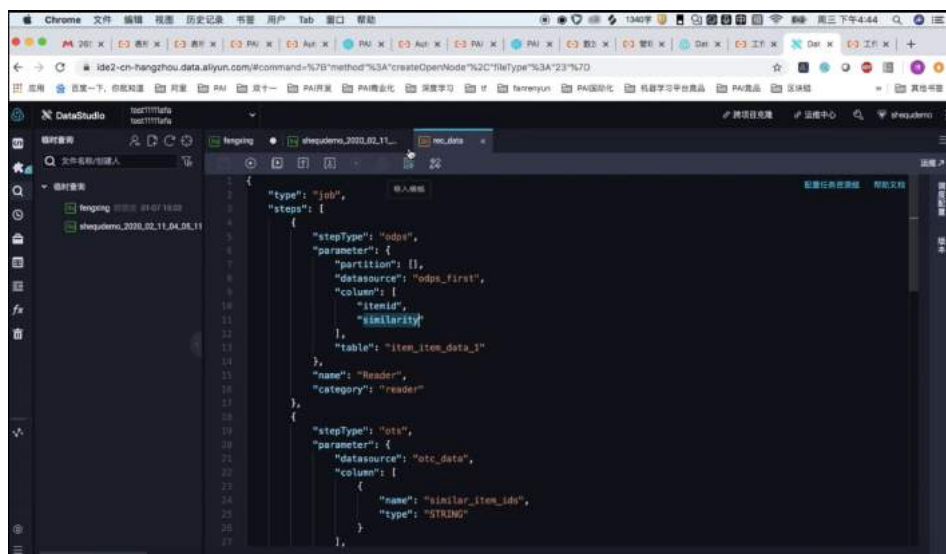
首先我们在 PAI-Studio 里基于你的原始数据生成你的协同过滤后的结果数据，这是两张表。



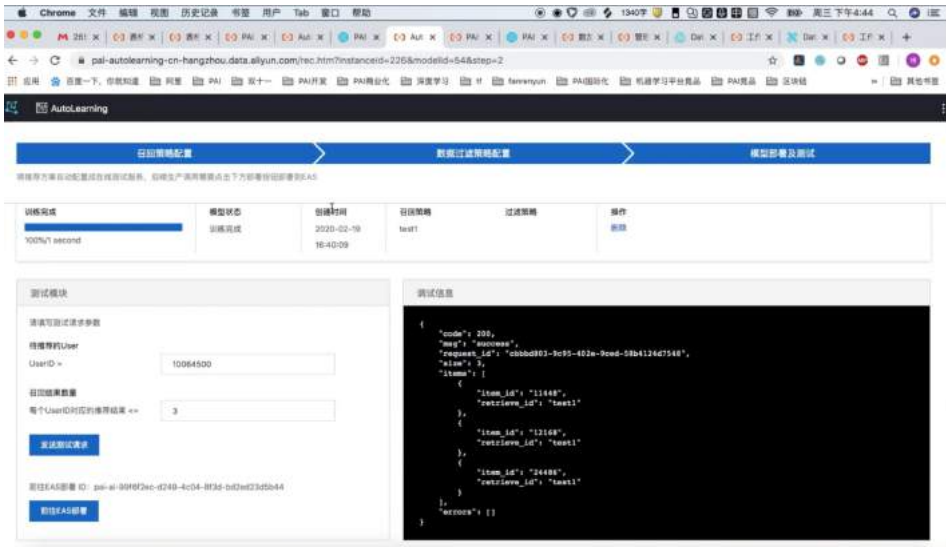
接着我们在 TableStore，即表格存储里去把这两张表按照要求的格式建立出来。



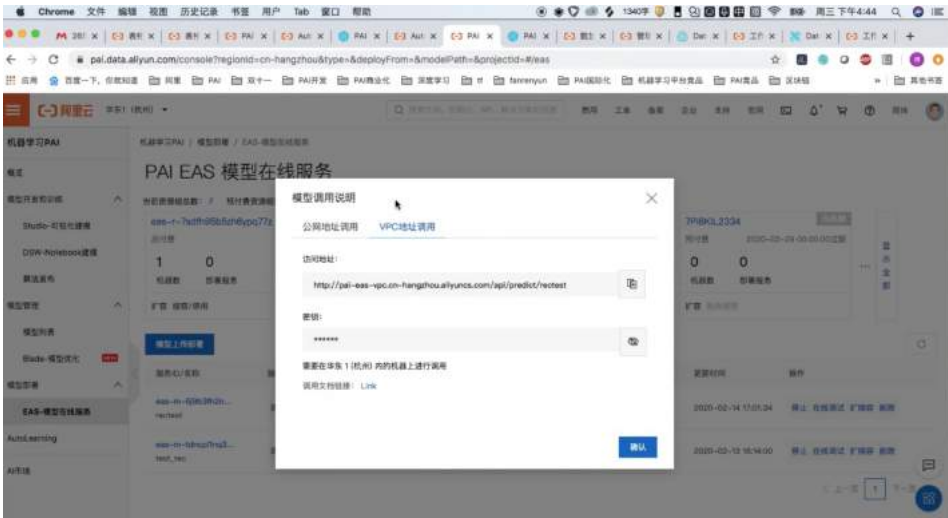
然后利用 Dataworks 去把这个数据从 PAI-Studio 里灌到表格存储里。



接着在 AutoLearning 里去把 TableStore 的两个数据配置出来，配出来呈一个策略。

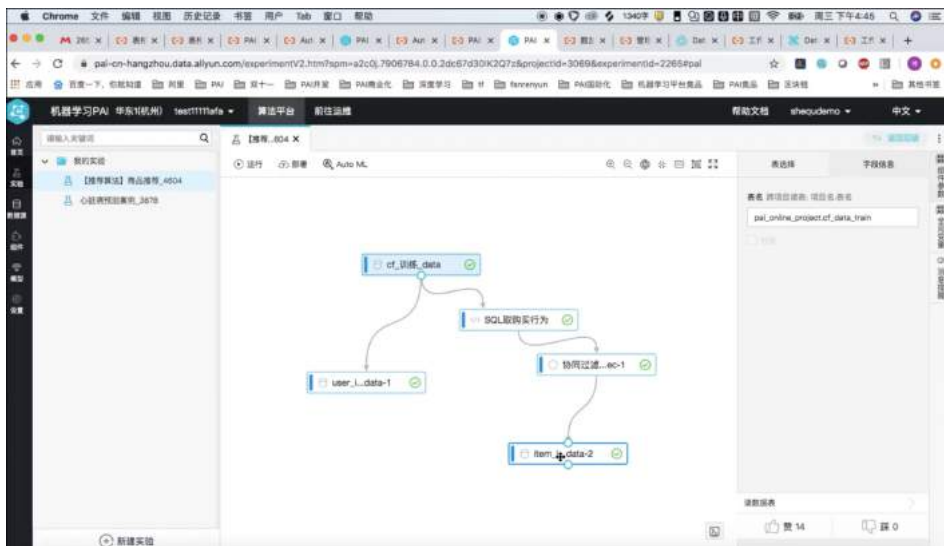


然后最后把这个策略变成一个 PAI-EAS 的服务，你可以在这看一个调用方式。最终用户拿到的结果就是这条服务。



如果想更新迭代里面的整个的数据，你就可以去改最原始的这张表。因为整套服务都可以做成一个自动化的流程，都可以利用 Dataworks 的数据调度系统去做成一个自动化的系统。所以你只要每天去更新这里的原始的数据就可以了，剩下的业务

都可以自动化。



参考资料

最后，介绍一下我们给大家准备的一些资料。这第一个 link 它对应的是 PAI 团队结合自身过去几年在推荐领域的一些探索，总结了 140 页的推荐业务的动手实践文档。没有机器学习背景的人基于我们这些文档，也可以在一周之内搭建一套企业级的推荐系统，大家如果感兴趣可以去用一下。另外这一个 PAI 的产品地址。

140 页推荐业务动手实践文档：

<https://www.aliyun.com/acts/best-practice/preview?&id=378791>

机器学习 PAI：

<https://www.aliyun.com/product/bigdata/product/learn>

PAI 可视化建模

训练常用机器学习算法及丰富的可视化组件，您无需代码基础，通过拖拉拽即可训练模型。

创建项目 返回名称 请输入

项目名称	显示名	训练模式	所属区域	项目管理员	MaxCompute资源	创建时间	开启GPU	操作
bert_test	bert_test	单灶 2 (北京)	shequdemo	bert_test		2020年5月11日 12:35	<input checked="" type="checkbox"/>	进入机器学习
erf_test	erf_test	单灶 2 (北京)	shequdemo	erf_test		2020年4月27日 14:29	<input type="checkbox"/>	进入机器学习
rec_test_yc	rec_test_yc	单灶 2 (北京)	shequdemo	rec_test_yc		2020年4月4日 22:35	<input type="checkbox"/>	进入机器学习
LogTextTest1...	LogTextTest1...	单灶 2 (北京)	shequdemo	LogTextTest1...		2020年3月11日 17:28	<input type="checkbox"/>	进入机器学习
test_yc3	test_yc3	单灶 2 (北京)	shequdemo	test_yc3		2019年12月23日 15:54	<input checked="" type="checkbox"/>	进入机器学习
wowai_test	wowai_test	单灶 2 (北京)	shequdemo	wowai_test		2019年12月20日 22:43	<input checked="" type="checkbox"/>	进入机器学习
taylor	taylor	单灶 2 (北京)	shequdemo	taylor		2019年10月8日 11:35	<input type="checkbox"/>	进入机器学习
Cartner	Cartner	单灶 2 (北京)	shequdemo	Cartner		2019年9月23日 14:00	<input checked="" type="checkbox"/>	进入机器学习
demo0010	demo0010	单灶 2 (北京)	shequdemo	demo0010		2019年8月18日 11:57	<input type="checkbox"/>	进入机器学习
lanch_comp...	lanch_comp...	单灶 2 (北京)	shequdemo	lanch_compell...		2019年8月20日 11:48	<input type="checkbox"/>	进入机器学习

共有19条，每页显示: 10 1 2 下一页 >



阿里云开发者“藏经阁”
海量免费电子书下载