上篇：推荐系统的实践与思考

人人都是产品经理
发布时间：07-25 15:27 | 深圳聚力创想信息科技有限公司

> 那么当我们真正要开始做一个推荐系统时，需要从哪几个方面考虑问题？不同方面需要注意什么？笔者分享了在实际情况中遇到的一些问题以及总结出的解决方法。

我之前进行过一个小调查，得知大家普遍在工作中遇到的与推荐系统相关的问题是：“数据太稀疏、数据没有形成闭环、数据没办法跟其他系统结合”等等，这些内容，是摆在我们面前的实际问题，那么当我们真正要开始做一个推荐系统时，需要从几方面考虑问题呢？

第一，**算法**。到底应该选择什么样的算法？无论是协同过滤还是其他算法，都要基于自己的业务产品；

第二，**数据**。当确定了算法时，应该选择什么样的数据？怎样加工数据？用什么样方法采集数据？有句话叫做“机器学习=模型+数据”，即便拥有了一个很复杂的模型，在数据出现问题的情况下，也无法在推荐系统里面发挥很好的效果；

第三，**在线服务**。当模型训练完毕，数据准备充分之后，就会面对接收用户请求返回推荐结果的事项，这其中包含两个问题。其一，返回响应要足够迅速。如果当一个用户请求后的一秒钟才返回推荐结果，用户很可能因丧失耐心而流失。其二，如何让推荐系统具有高可扩展性。当 DAU 从最初的十万涨到一二百万时，推荐系统还能像最初那样很好地挡住大体量的请求吗？这都是在线服务方面需要考虑和面临的问题；

第四，**评估效果**。做好上述三点，并不代表万事大吉，一方面，我们要持续迭代推荐算法模型与结构，另一方面要去构建一套比较完整、系统的评价体系和评估方法，去分析推荐效果的现状以及后续的发展。

作者最新文章

上篇：推荐系统的实践与思考

价值与风险，产品经理的核心思考

阿里云、腾讯云的恩怨情仇

相关文章

详解互联网运营的本质与底层逻辑

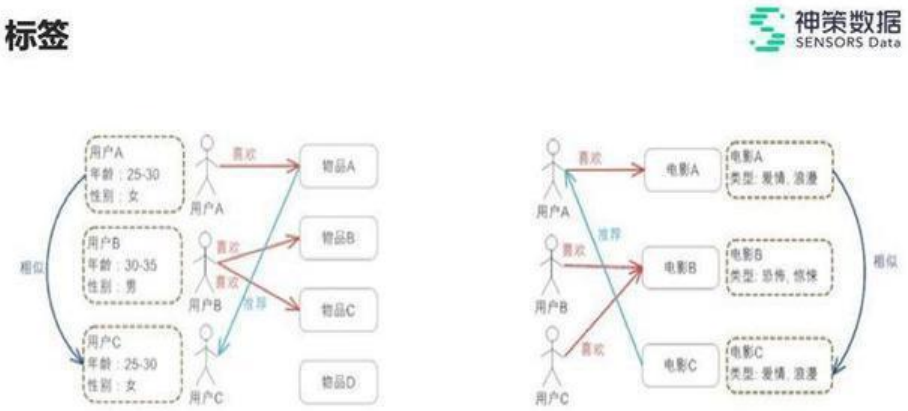
私域流量时代：中国所有 to C 生意都值得重做

会员运营（1）：会员的三种基本模式

5步高效解决运营疑难杂症

01 算 法

在各种算法中，大家最容易想到的就是一种基于标签的方法。



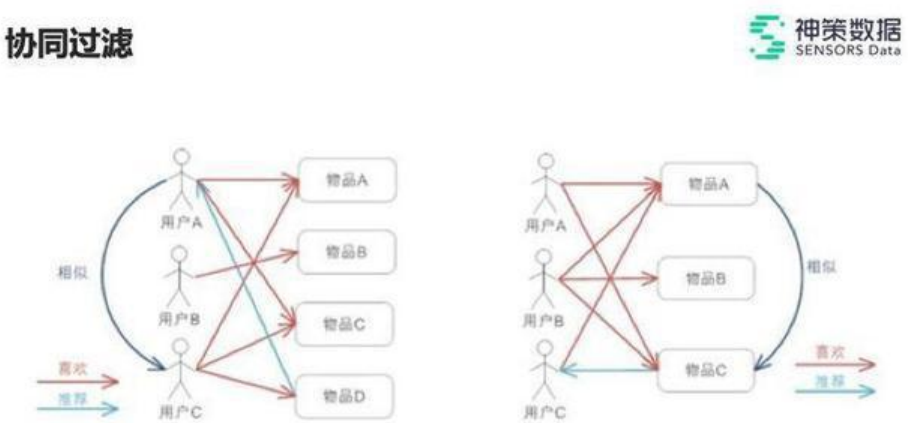
如上图所示，标签可分为两种：

第一种，用户标签。假设我们拥有一部分用户标签，知道每一个用户的年龄、性别等信息，当某类年龄和某种性别的用户喜欢过某一个物品时，我们就可以把该物品推荐给具有同样年龄、性别等用户标签的其他用户；

第二种，内容标签。与用户标签的思路相似，如果用户喜欢过带有内容标签的物品，我们就可以为他推荐具有同样标签的内容。

但很明显，这种基于标签的方法有一个重要的缺点——它需要足够丰富的标签。也许在多产品中，可能并没有标签或者标签数量非常稀疏，所以标签的方法显然不足以应对。

另外，协同过滤也是一种非常经典、被较多人提及的一种方法，是一种常见且有效的思路。



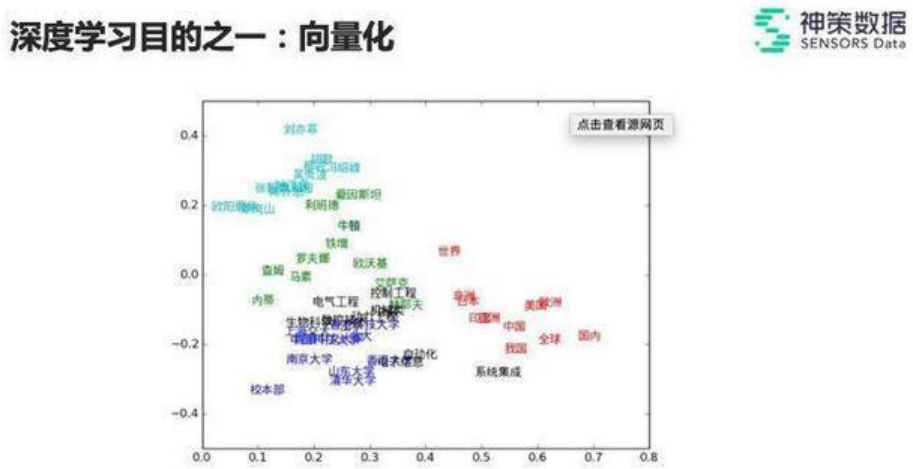
运营：要有化繁为简的能力

解决推荐的问题，利用用户的行为数据去构建推荐算法。

1. 深度学习的目的之一：向量化

推荐系统其实是在做一个关于“匹配”的事情，把人和物做匹配。看似很难的推荐系统，其实也有简单的思路——做人和物的匹配，把该用户可能感兴趣的物品推荐给他（她）。如果站在数学的角度去思考这个问题，我们如何去计算人和物的相似度匹配呢?

在推荐领域，深度学习的目的之一就是尝试将人和物向量化，即把某个人和某个物品学习成一种统一的表示方式，随后在这个统一的表示方式中计算这个人和物品的相似度，当人和物都映射到同一个可比较的空间中时，就能够基于计算结果去执行相关的内容推荐。



把最终的结果映射到这张二维的平面图表里，用户认为相似的内容就会映射在向量上，当拥有内容向量之后，之后再将用户映射进来即可，比如用户到达了上图某个地方，根据他所处的位置，可以向其推送教育、娱乐、科学、地理等内容。

讲到这里，有些朋友就会提出疑问：既然深度学习如此复杂，那在实践中究竟有没有作用？其实站在实战经验的角度来看，当具备一定的数据量时，会带来比较明显的效果提升，但当你要去搭建一个深度学习模型的时候，可能真的会遇到很多问题。比如：

用多少数据量去训练模型是可以的？训练数据该用什么格式？多“深”才算深度模型？训练模型太慢了怎么办等。这些关于在搭建深度学习模型时遇到的困难与解决方法，会在以后跟大家分享。

2. 冷启动

冷启动是算法部分经常遇到的问题，在冷启动阶段，数据比较稀疏，很难利用用户的行为数据实现个性化推荐。冷启动的问题分为两种：新内容的冷启动、新用户的冷启动。接下来，我们分享一下新内容的冷启动要如何实现。

举个例子，资讯场景的需求往往是将发布的新内容（如 10 分钟内发布的内容），以实时且个性化的方式分发到用户的推荐结果中去。

上图这篇文章在 17:41 发出，那么就需要在极短的时间内根据这篇文章的内容去做一些个性化的相关推荐。此文内容围绕美食展开，用户点开这篇文章之后，文章的相关推荐里面就要有跟美食相关的一些内容。当我们要在如此实时的环境中实现推荐效果的话，其实没办法去依赖用户的行为。

这时，我们尝试提供一种思路，一种基于深度学习的语义理解模型。

这个模型跟我们前面分享的内容有一个很大的区别就是——不需要用户行为，只需要分析用户文本，基于用户的内容去给每一篇文章生成一个向量。这和前面提到的模型也有相通的地方：第一，用深度学习的思路去解决问题；第二，用向量化的思路解决问题。我们只需要训练出文章的语义向量，获得文章与文章之间的相似度，从而得知文章和用户之间的相关性。

3. 召回、排序、规则

如今的推荐系统已经做得相当复杂，特别是在一些大规模的应用场景中，比如说今日头条的 Feed 流，淘宝的“猜你喜欢”等，都拥有一个非常复杂的推荐系统，这个推荐系统中的各个模块可能会涉及到很多的实验算法，在一个系统中，出现 10 个或者 20 个模型都很常见。那么怎么把这些模型有效地融合成一个真正的系统呢?

（1） 召回

召回，即从海量的内容里去召回每一个用户他可能感兴趣的内容，前提是——拥有海量的内容，因为当内容不足时，也就不需要去搭建复杂的推荐系统。所以，当有海量 Item 时，需要用召回的算法从不同类别的内容里为用户生成他可能感兴趣的内容。

比如某位用户既喜欢体育内容，也喜欢军事内容，那么在第一步，无论用哪些模型，都希望达到为该用户生成一些体育、军事相关内容的效果。另一个用户可能喜欢美食和游戏，在召回阶段，我们就希望通过模型去为他生成一些美食和游戏相关的内容。

在召回阶段可能就会存在许多个模型。而经过召回阶段之后，尽管生成的是该用户可能感兴趣的内容，但这些内容实际并没有融合到一起，是一种乱序的状态。

（2） 排序

排序，即将召回出来的内容做统一排序。排序过程其实就是给每部分内容打分的过程，预测每一个用户对每一部分内容的感兴趣程度，从而获知每一个用户对每部分内容的偏好程度。

（3） 规则

系，所以有些常见的业务需求要通过规则去实现。

举个例子，部分推荐场景中会出现一些运营精选的内容，运营同事的需求是：保证每十条内容中都有一条编辑精选内容，而这个需求，只能通过规则实现，而不是通过算法。

一个比较复杂的推荐系统通常分为召回、排序、规则这三个步骤。首先召回用户感兴趣的内容，第二为用户生成一个排序列表，第三用规则解决一些产品、运营方面提出的需求。

02 数 据

总是会听到一个这样的说法，“推荐算法的效果是由模型与数据所决定的”，即模型只占推荐效果中的一部分，另外一个非常重要的部分就是数据。那么我们究竟需要哪些数据？在一个实际的推荐系统中，哪些数据是有可能发挥作用的？我们又能拿到哪些数据？

通常来说一般会有四类数据：用户行为、物品信息、用户画像以及外部数据。

1. 用户行为

用户行为数据最为重要，几乎没有哪一个推荐系统可以直接表示不需要用户行为数据。一方面，用户行为数据是训练模型中的一个重要数据来源，另外一方面，需要通过用户的行为反馈，技术同事才能知道推荐系统到底做得如何。搭建推荐系统的一个秘籍就是积累用户行为数据，如果没有将重要的用户行为做采集，例如在电商场景中，如果只是记录最终的下单数据，那么离推荐系统的数据要求还是有一定的距离。

2. 物品信息

物品信息指推荐系统中能采集到的描述每一个内容的信息。以电商场景为例，在录入一件具体物品时，录入商品的品牌、价格、品类、上架时间等就是我们要收集的物品信息。假设在电商场景中，如果并不清楚每个商品的品牌，也就无法从一些物体的描述信息中去提取某个商品到底属于何种品牌，那么推荐效果自然受到限制。当物品信息采集的足够丰富时，对推荐系统的效果就会有一定的帮助。

3. 用户画像

在传统的思路中，认为用户画像里面存储的实际还是用户的标签，但在很多实际场景中标签数量少、维度粗，可能根本不具备去给用户打标签的能力，这种传统的“标签式”想法，就会限制搭建推荐系统的思路。

而从深度学习的角度出发，用户画像中储存的并不是通常理解的“标签”，他可能存储的是这个人的向量，深度学习是把人和物品做向量化，但这个向量是不可被理解的，即我们可能并不知道这个向量表示的是什么意思，当我们看到某个用户对应的向量，我们也不知道他是对体育、音乐或是娱乐感兴趣，但我们仍能够通过向量去为他推荐其感兴趣的内容。

4. 外部数据

有的人会迷信外部数据，觉得自己的数据量不够，所以一定要去购买阿里或者是腾讯的外部数据来充实用户画像，从而提高推荐系统的效果。甚至有人认为推荐系统效果不好，是因为没有外部数据。

首先，要先验证自己的这批用户群跟所购买的外部数据能发生多少交集。假如一个游戏平台，购买了阿里的外部数据，而这样的外部数据可能只能告诉你用户到底是喜欢买衣服、买车还是买电子产品，这样的信息对游戏平台有用吗？

假设购买的外部数据恰好命中了业务场景，可能会发挥一定的作用，但实际上，能够同时命中用户群体和标签的情况也并不常见。

大家不要认为上述的 4 种数据比较容易理解，所以获取时也会比较简单。其实我和我们神策团队在去构建一个实际的推荐系统时，消耗我们人力的地方往往不是算法，反而是怎么去得到正确的数据，接下来我们以用户行为数据为例，与大家分享应该如何获取我们所需要的用户行为数据？

这时候我们就要思考，当我们想去获取用户行为数据时，到底希望用户行为数据能给我们带来什么样的作用？

我总结为以下几个方面：

1. 我们希望用户行为数据能用来训练模型，这是非常重要的一个方面。比如我给某个用户推荐十件商品，其中有两件商品发生了点击行为，模型中就会觉得这两条数据是正例，其他是负例。所以，我们需要用户行为数据作为模型的训练数据；
2. 我们希望用户行为数据能够验证效果。推荐系统上线之后，需要用户行为数据来反馈推荐到底做得怎么样。比如点击率上升说明效果变好，点击率下降、负反馈变多、用户流失，说明推荐系统可能出现了问题；
3. 我们希望用户行为数据能够支持我们去看 A/B Test 效果。模型上线一定要基于 A/B Test，我们需要知道此次上线到底比之前的推荐算法、推荐系统等效果如何。这样，我们才能判断这一次的迭代是否有效，如果有效就将其全量，如果无效，则进一步迭代；
4. 我们希望它能够帮助我们分析问题。我们将推荐系统上线之后，可能会碰到一些懊恼的问题，比如点击率并没有发生变化，甚至效果变差，毕竟不可能每一次迭代的效果都是上升的，所以我们希望行为数据能够定位到此次推荐系统上线后效果不理想的原因。如果上线后效果不错，此时我们希望行为数据能够分析到底是哪些因素使效果变好。

那我们应该如何去获取满足我们这些需求的行为数据呢？以曝光日志中的第一个字段 exp_id 为例，exp_id 的中文的意思是实验 ID。

前面提到了我们希望用户行为数据是能支持 A/B Test 的，那么如何知道每一条数据是来自哪一组实验呢？此时，我们需要一个 exp_id 字段去记录每一条曝光日志是来自哪组实验。当我们再次分析 A/B Test 效果时，就可以根据一个 exp_id 字段去区分不同实验所带来的曝光和点击。

内容曝光	exp_id	实验ID
	strategy_id	策略ID
	retrieve_id	召回来源ID
	log_id	服务追踪ID
	contentType	内容类型
	contentID	内容ID
	contentTitle	内容标题
	contentChannel	内容频道
	contentTag	内容标签
	operationTag	运营标签
	exposureFrom	曝光来源
	articleSource	文章来源
	relatedArticleID	直接关联的文章ID
	showPosition	展现位置
	photoCounts	展现图片数量
	showUpTime	展示时间
	articlePublishTime	文章发布时间

在曝光日志中我们常常讨论如何设计一些常用字段，而另外一个具体问题就是——我们怎么去采集这些数据?

简单来说，当用户在产品里面发生一些用户行为，怎么把这个数据最终落到服务器的日志中，从而用于模型训练和效果分析呢?

做用户行为的采集，通常有两种方式。第一种就是自助埋点，客户端先把用户的行为记录下来，之后传给服务端，服务端再去传给推荐引擎。另外一个埋点方式是 SDK 埋点，我们直接使用 SDK 去做推荐引擎的埋点。

SDK 埋点有两个方面的优势：

1. SDK 埋点的接入成本低，它有比较成熟的埋点事件和埋点验证方案。另外 SDK 有埋点接口和文档指导客户埋点，无需关注上报问题；
2. SDK 埋点的容错性比较高。如果是自助埋点，从客户端到推荐引擎经过了服务端，数据出现问题，难以回溯埋点问题、传输问题、数据质量维护成本高，SDK 埋点就会相对方便。

那么当有了行为数据之后，如何去训练模型？通常会有以下几个步骤：

1. 构造正负例。比如给用户推荐十条商品，有几条发生点击，就有几条正例，其他没有发生点击就是负例；
2. 构造特征工程。稍后会以一个电商场景为例，具体讲解通常情况下，如何构造特征工程；
3. 数据采样。数据采样对整个模型训练的效果影响较大。

下面以电商场景为例，讲解如何做特征工程，主要分为 2 个方面：

1. 商品维度。在商品的维度里，我们可能关注一些商品的品类、品牌、价格、所面向的性别，以及各种用户行为反馈的一些数据，比如点击率、收藏比率等，这些内容一方面体现了商品本身的一些属性，同时还体现商品的质量；
2. 用户层面，通常首先考虑用户的年龄和性别。因为在电商领域中男性所偏重的商品和女性之间存在较大差异。另外还有用户的品类偏好、品牌偏好，以及价格偏好等。

在数据方面，跟大家分享一下我在实际工作中遇到的“坑”：某一次小的流量上线之后，我和团队成员发现效果不如预期，根据以往的实践经验来说，不应该是这么差的

第一，命中行为模型的用户较少，通常情况下，只要不是一个新用户，理论上来说，都应该能够命中我的行为模型。我们当时的新用户比例在 20% 以下，而命中模型的用户大概仅为 30%，说明大量的用户没有命中到模型；

第二，很多请求的 ID 未出现在日志中，当时我们怀疑，是否我们的推荐结果被别人作弊刷掉了，因为用“作弊”能很好地解释这些请求并未落到日志中的原因。

但最终，我们发现并不是作弊的问题，而是因为用户 ID 没有统一。前端在用他们理解的一套用户 ID 体系打日志，但是后端在用另外一套用户 ID 体系发送请求，于是所有的数据无法对上，后端过来的请求总是新用户，而训练出来的模型命中不了任何用户，最终，我们建立了一系列的方法和工具以及流程去保证整个用户 ID 体系的一致性。

由于篇幅限制，“在线服务”和“效果评估”将在下一篇文章进行介绍，希望对你有帮助！

本文由 @研如玉 原创发布于人人都是产品经理，未经许可，禁止转载

题图来自Unsplash，基于CC0协议