

26 | 在线测试：如何在推荐服务器内部实现A/B 测试？

你好，我是王喆。这节课我们来聊一聊推荐系统的线上 A/B 测试。

上两节课，我们进行了推荐系统离线评估方法和指标的学习。但是无论采用哪种方法，离线评估终究无法还原线上的所有变量。比如说，视频网站最想要提高的指标是用户观看时长，在离线评估的环境下可以模拟出这个指标吗？显然是非常困难的。即使能够在离线环境下生成这样一个指标，它是否能真实客观地反映线上效果，这也要打一个问号。

所以，对于几乎所有的互联网公司来说，线上 A/B 测试都是验证新模型、新功能、新产品是否能够提升效果的主要测试方法。这节课，我们就来讲一讲线上 A/B 测试，希望通过今天的课程，能帮助你了解到 A/B 测试的基本原理，A/B 测试的分层和分桶方法，以及怎么在 SparrowRecSys 的推荐服务器中实现 A/B 测试模块。

如何理解 A/B 测试？

A/B 测试又被称为“分流测试”或“分桶测试”，它通过把被测对象随机分成 A、B 两组，分别对它们进行对照测试的方法得出实验结论。具体到推荐模型测试的场景下，它的流程是这样的：先将用户随机分成实验组和对照组，然后给实验组的用户施以新模型，给对照组的用户施以旧模型，再经过一定时间的测试后，计算出实验组和对照组各项线上评估指标，来比较新旧模型的效果差异，最后挑选出效果更好的推荐模型。

好了，现在我们知道了什么是线上 A/B 测试。那它到底有什么优势，让几乎所有的互联网公司主要使用它来确定模型最终的效果呢？你有想过这是什么原因吗？我总结了一下，主要有三点原因。接下来，我们就一起来聊聊。

首先，离线评估无法完全还原线上的工程环境。 一般来讲，离线评估往往不考虑线上环境的延迟、数据丢失、标签数据缺失等情况，或者说很难还原线上环境的这些细节。因此，离线评估环境只能说是理想状态下的工程环境，得出的评估结果存在一定的失真现象。

其次，线上系统的某些商业指标在离线评估中无法计算。 离线评估一般是针对模型本身进行评估的，无法直接获得与模型相关的其他指标，特别是商业指标。像我们上节课讲的，离线评估关注的往往是 ROC 曲线、PR 曲线的改进，而线上评估却可以全面了解推荐模型带来的用户点击率、留存时长、PV 访问量这些指标的变化。

其实，这些指标才是最重要的商业指标，跟公司要达成的商业目标紧密相关，而它们都要由 A/B 测试进行更全面准确的评估。

最后是离线评估无法完全消除数据有偏（Data Bias）现象的影响。 什么叫“数据有偏”呢？因为离线数据都是系统利用当前算法生成的数据，因此这些数据本身就不是完全客观中立的，它是用户在当前模型下的反馈。所以说，用户本身有可能已经被当前的模型“带跑偏了”，你再用这些有偏的数据来衡量你的新模型，得到的结果就可能不客观。

正是因为离线评估存在这三点硬伤，所以我们必须利用线上 A/B 测试来确定模型的最终效果。明确了这一点，是不是让我们的学习更有方

向了？接下来，我们再深入去学习一下 A/B 测试的核心原则和评估指标。

A/B 测试的“分桶”和“分层”原则

刚才，我们说 A/B 测试的原理就是把用户分桶后进行对照测试。这看上去好像没什么难的，但其实我们要考虑的细节还有很多，比如到底怎样才能对用户进行一个公平公正的分桶呢？如果有多组实验在同时做 A/B 测试，怎样做才能让它们互不干扰？

下面，我就来详细的讲一讲 A/B 测试的“分桶”和“分层”的原则，告诉你让 A/B 测试公平且高效的执行方法长什么样。

在 A/B 测试分桶的过程中，我们需要注意的是**样本的独立性和分桶过程的无偏性**。这里的“独立性”指的是同一个用户在测试的全程只能被分到同一个桶中。“无偏性”指的是在分桶过程中用户被分到哪个实验桶中应该是一个纯随机的过程。

举个简单的例子，我们把用户 ID 是奇数的用户分到对照组，把用户 ID 是偶数的用户分到实验组，这个策略只有在用户 ID 完全是随机生成的前提下才能说是无偏的，如果用户 ID 的奇偶分布不均，我们就无法保证分桶过程的无偏性。所以在实践的时候，我们经常会使用一些比较复杂的 Hash 函数，让用户 ID 尽量随机地映射到不同的桶中。

说完了分桶，那什么是分层呢？要知道，在实际的 A/B 测试场景下，同一个网站或应用往往要同时进行多组不同类型的 A/B 测试。比如，前端组正在进行不同 App 界面的 A/B 测试的时候，后端组也在进行不同中间件效率的 A/B 测试，同时算法组还在进行推荐场景 1 和推荐场景 2 的 A/B 测试。这个时候问题就来了，这么多 A/B 测试同时进行，我们怎么才能让它们互相不干扰呢？

你可能会说，这还不简单，我们全都并行地做这些实验，用户都不重叠不就行了。这样做当然可以，但非常低效。你如果在工作中进行过 A/B 测试的话肯定会知道，线上测试资源是非常紧张的，如果不进行合理的设计，很快所有流量资源都会被 A/B 测试占满。

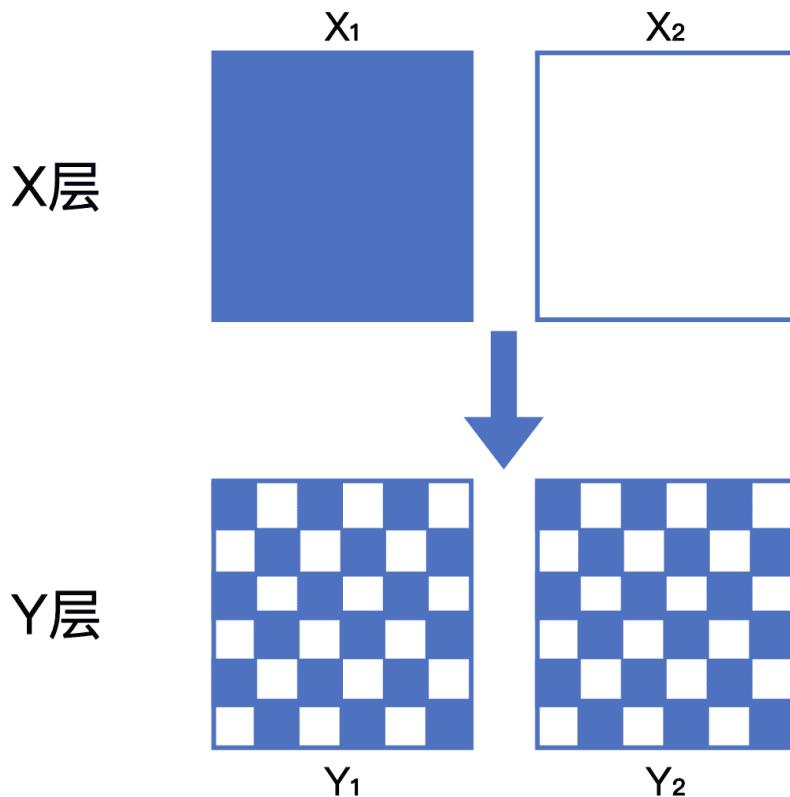
为了解决这个问题，我们就要用到 A/B 测试的分层原则了。Google 在一篇关于实验测试平台的论文《Overlapping Experiment Infrastructure: More, Better, Faster Experimentation》中，详细介绍了 A/B 测试分层以及层内分桶的原则。

如果你没看过这篇论文，没关系，你记住我总结出来的这句话就够了：**层与层之间的流量“正交”，同层之间的流量“互斥”**。它是什么意思呢？接下来，我就针对这句话做个详细的解释。

首先，我们来看层与层之间的流量“正交”，它指的是层与层之间的独立实验的流量是正交的，一批实验用的流量穿越每层实验时，都会再次随机打散，然后再用于下一层的实验。

这么说好像还是太抽象，我们来看下面的示意图。假设，在一个 X 层的实验中，流量被随机平均分为 X1（蓝色）和 X2（白色）两部分。当它们穿越到 Y 层的实验之后，X1 和 X2 的流量会被随机且均匀地分配给 Y 层的两个桶 Y1 和 Y2。

如果 Y1 和 Y2 的 X 层流量分配不均匀，那么 Y 层的样本就是有偏的，Y 层的实验结果就会被 X 层的实验影响，也就无法客观地反映 Y 层实验组和对照组变量的影响。



层与层之间流量正交示例

理解了第一句话，我们再来看看什么叫同层之间的流量“互斥”。这里的“互斥”具体有 2 层含义：

1. 如果同层之间进行多组 A/B 测试，不同测试之间的流量不可以重叠，这是第一个“互斥”；
2. 一组 A/B 测试中实验组和对照组的流量是不重叠的，这是第二个“互斥”。

在基于用户的 A/B 测试中，“互斥”的含义可以被进一步解读为，不同实验之间以及 A/B 测试的实验组和对照组之间的用户是不重叠的。特别是对推荐系统来说，用户体验的一致性是非常重要的。也就是说我们不可以让同一个用户在不同的实验组之间来回“跳跃”，这样会严重

损害用户的实际体验，也会让不同组的实验结果相互影响。因此在 A/B 测试中，保证同一用户始终分配到同一个组是非常有必要的。

A/B 测试的“正交”与“互斥”原则共同保证了 A/B 测试指标的客观性，而且由于分层的存在，也让功能无关的 A/B 测试可以在不同的层上执行，充分利用了流量资源。

在清楚了 A/B 测试的方法之后，我们要解决的下一个问题是，怎么选取线上 A/B 测试的指标。

线上 A/B 测试的评估指标

一般来说，A/B 测试是模型上线前的最后一道测试，通过 A/B 测试检验的模型会直接服务于线上用户，来完成公司的商业目标。因此，**A/B 测试的指标应该与线上业务的核心指标保持一致**。这就需要我们因地制宜地制定最合适的推荐指标了。

具体怎么做呢？实际也不难，那在实际的工作中，我们需要跟产品、运营团队多沟通，在测试开始之前一起制定大家都认可的评估指标。为了方便你参考，我在下表中列出了电商类推荐模型、新闻类推荐模型、视频类推荐模型的主要线上 A/B 测试评估指标，你可以看一看。

推荐系统类别	线上A/B测试评估指标	离线评估指标
电商类推荐模型	点击率、转化率、客单价（用户平均消费金额）	准确率、精确率、召回率、F1-score、ROC AUC、PR AUC等
新闻类推荐模型	留存率（x日后仍活跃的用户数/x日前的用户数）、平均停留时长、平均点击个数	
视频类推荐模型	播放完成率（播放时长/视频时长）、平均播放时长、播放总时长	



看了这些指标，我想你也发现了，线上 A/B 测试的指标和离线评估的指标（诸如 AUC、F1-score 等），它们之间的差异非常大。这主要是因为，离线评估不具备直接计算业务核心指标的条件，因此退而求其次，选择了偏向于技术评估的模型相关指标，但公司更关心的是能够驱动业务发展的核心指标，这也是 A/B 测试评估指标的选取原则。

总的来说，在具备线上环境条件时，利用 A/B 测试验证模型对业务核心指标的提升效果非常有必要。从这个意义上讲，线上 A/B 测试的作用是离线评估永远无法替代的。

SparrowRecSys 中 A/B 测试的实现方法

搞清楚了 A/B 测试的主要方法，下一步就让我们一起在 SparrowRecSys 上实现一个 A/B 测试模块，彻底掌握它吧！

既然是线上测试，那我们肯定需要在推荐服务器内部来实现这个 A/B 测试的模块。模块的基本框架不难实现，就是针对不同的 userId，随机分配给不同的实验桶，每个桶对应着不同的实验设置。

比较方便的是，我们可以直接在上一篇刚实现过的“猜你喜欢”功能上进行实验。实验组的设置是算法 NerualCF，对照组的设置是 Item2vec Embedding 算法。接下来，我们说一下详细的实现步骤。

首先，我们在 SparrowRecSys 里面建立了一个 ABTest 模块，它负责为每个用户分配实验设置。其中，A 组使用的模型 bucketAModel 是 emb，代表着 Item2vec Embedding 算法，B 组使用的模型 bucketBModel 是 Nerualcf。除此之外，我们还给不在 A/B 测试的用户设置了默认模型 emb，默认模型是不在实验范围内的用户的设置。

模型设置完，就到了分配实验组的阶段。这里，我们使用 getConfigByUserId 函数来确定用户所在的实验组。具体怎么做呢？因为这个函数只接收 userId 作为唯一的输入参数，所以我们利用 userId 的 hashCode 把数值型的 ID 打散，然后利用 userId 的 hashCode 和 trafficSplitNumber 这个参数进行取余数的操作，根据余数的值来确定 userId 在哪一个实验组里。

你可能对 trafficSplitNumber 这个参数的作用还不熟悉，我来进一步解释一下。这个参数的含义是把我们的全部用户分成几份。这里，我们把所有用户分成了 5 份，让第 1 份用户参与 A 组实验，第 2 份用户参与 B 组实验，其余用户继续使用系统的默认设置。这样的操作就是分流操作，也就是把流量划分之后，选取一部分参与 A/B 测试。

```
public class ABTest {  
    final static int trafficSplitNumber = 5;  
    final static String bucketAModel = "emb";  
  
    final static String bucketBModel = "nerualcf";  
    final static String defaultModel = "emb";  
    public static String getConfigByUserId(String  
        if (null == userId || userId.isEmpty()) {  
            return defaultModel;  
        }  
        if (userId.hashCode() % trafficSplitNumber == 0) {  
            System.out.println(userId + " is in bucketA");  
            return bucketAModel;  
        } else if (userId.hashCode() % trafficSplitNumber == 1) {  
            System.out.println(userId + " is in bucketB");  
            return bucketBModel;  
        } else {  
            System.out.println(userId + " isn't in either bucket");  
            return defaultModel;  
        }  
    }
```

```
    }  
}
```

上面是 A/B 测试模块的主要实现。在实际要进行 A/B 测试的业务逻辑中，我们需要调用 A/B 测试模块来获得正确的实验设置。比如，我们这次选用了猜你喜欢这个功能进行 A/B 测试，就需要在相应的实现 RecForYoService 类中添加 A/B 测试的代码，具体的实现如下：

```
if (Config.IS_ENABLE_AB_TEST){  
    model = ABTest.getConfigById(userId);  
}  
//a simple method, just fetch all the movie in the  
List<Movie> movies = RecForYouProcess.getRecList(I
```

我们可以看到，这里的实现非常简单，就是调用 ABTest.getConfigById 函数获取用户对应的实验设置，然后把得到的参数 model 传入后续的业务逻辑代码。需要注意的是，我设置了一个全局的 A/B 测试使能标识 Config.IS_ENABLE_AB_TEST，你在测试这部分代码的时候，要把这个使能标识改为 true。

上面就是经典的 A/B 测试核心代码的实现。在实际的应用中，A/B 测试的实现当然要更复杂一些。比如，不同实验的设置往往是存储在数据库中的，需要我们从数据库中拿到它。再比如，为了保证分组时的随机性，我们往往会创建一些复杂的 hashCode 函数，保证能够均匀地把用户分到不同的实验桶中。但整个 A/B 测试的核心逻辑没有变化，你完全可以参考我们今天的实现过程。

小结

这节课，我们讲解了线上 A/B 测试的基本原理和评估指标，并且在 SparrowRecsys 上实现了 A/B 测试的模块。我带你从 A/B 测试的定义和优势、设计原则以及在线评估指标这三个方面回顾一下。

A/B 测试又叫“分流测试”或“分桶测试”，它把被测对象随机分成 A、B 两组，通过对照测试的方法得出实验结论。相比于离线评估，A/B 测试有三个优势：

1. 实验环境就是线上的真实生产环境；
2. 可以直接得到线上的商业指标；
3. 不受离线数据“数据有偏”现象的影响。

在 A/B 测试的设计过程中，我们要遵循层与层之间的流量“正交”，同层之间的流量“互斥”这一设计原则，这样才能既正确又高效地同时完成多组 A/B 测试。除此之外，在线上评估指标的制定过程中，我们要尽量保证这些指标与线上业务的核心指标保持一致，这样才能更加准确地衡量模型的改进，有没有帮助到公司的业务发展，是否达成了公司的商业目标。

为了方便你复习，我把一些核心的知识点总结在了表格中，你可以看一看。

知识点	关键描述
A/B测试定义	“分流测试”或“分桶测试”，它通过把被测对象随机分成A、B两组，通过对照测试的方法得出实验结论
线上测试的优势	1. 实验环境就是线上的真实产生环境； 2. 可以直接得到线上的商业指标； 3. 不受离线数据“数据有偏”现象的影响
A/B测试的设计原则	层与层之间的流量“正交”，同层之间的流量“互斥”
A/B测试的评估指标	与线上业务的核心指标保持一致，典型的包括点击率、留存率、播放时长等



课后思考

今天讲的 A/B 测试的分层和分桶的原则你都理解了吗？如果我们在测试模型的时候，一个实验是在首页测试新的推荐模型，另一个实验是在内容页测试新的推荐模型，你觉得这两个实验应该放在同一层，还是可以放在不同的层呢？为什么？

期待在留言区看到你的思考，如果有其他疑问也欢迎你随时提出来，我会一一解答，我们下节课见！