第一部分

基础

第 1 章 推荐系统概要

第 2 章 评级指标与实验机制

不论是做推荐、搜索、广告、还是其他互联网的应用，都需要用量化指标评价模型和策略带来的收益，整个系统的迭代优化都是围绕量化评价指标进行的。如果目标设定错了，那么一切的努力都是徒劳的。算法工程师和产品经理都应当将各种指标的定义和数值烂熟于心，只有真正理解评价指标，才能判断自己和他人的工作是否有价值。推荐系统算法工程师的日常工作就是设计和改进模型和策略，不断地做小流量 A/B 测试判断各种指标是否获得提升。本章详细详细讲解推荐系统的评价指标、以及 A/B 测试的机制。

2.1 推荐系统的评价指标

所有的信息流推荐系统都以用户规模、用户留存、消费作为核心指标（也叫北极星指标）。[1] 小红书、抖音这样的用户生成内容（user generated content，UGC）平台还把用户发布作为核心指标；腾讯视频这样的专业生产内容（professional generated content，PGC）平台花钱采购内容，而不考虑用户发布指标。用户发布指标由物品冷启动负责，留到相应的章节讨论。推荐系统还有很多非核心指标，比如点击率、交互率，这些指标只作为观测，而不作为决定是否允许新模型、新策略推全的依据。当然，如果一个策略严重损害点击率、交互率这样的非核心指标，那么久需要严格考察指标的兑换关系。

2.1.1 用户规模

推荐系统最重要的优化目标是吸引到更多用户、留住用户、让用户更活跃。业界常用日活用户数（daily active user，DAU）来衡量用户粘性、用户活跃程度。如果一位用户今天使用 1 次 APP、或者使用 10 次 APP，他对 DAU 的贡献都是 1。月活用户数（monthly active user，MAU）也是重要的观测指标。如果一个用户在未来 28 天内使用 1 次 APP、或者使用 10 次 APP，他对今天的 MAU 的贡献都是 1。周活用户数（weekly active user，WAU）的定义也是类似的。

小红书 APP 主要有搜索和推荐两个功能，对于这样的 APP，需要用推荐日活用户数（feed daily active user，FDAU）评价推荐产品做得好不好，用搜索日活用户数（search daily active user，SDAU）评价搜索产品做得好不好，而不能简单地用 APP 的整体 DAU。某些用户上小红书只做搜索，找到想要的信息就离开，不使用推荐；而某些用户只看推荐信息流，不使用搜索。因为很多用户既用搜索也用推荐，所以 SDAU 和 FDAU 都小于 DAU，而 SDAU+FDAU 则大于 DAU。

可以用 FDAU 与 DAU 的比值评价推荐产品的好坏，这个比值叫做推荐渗透率，即今天有多大比例的 APP 用户使用了推荐产品。比值越高，说明推荐做得越好，用户越习惯使用小红书的推荐功能。举个例子，一位用户原本登录小红书 APP 只是为了搜索出行攻略，但是小红书推荐做得好，吸引了用户的注意力，用户就点了进去，提升了 FDAU/DAU。

[1] 电商推荐系统的核心指标为营收，与信息流推荐系统有本质区别

FDAU/DAU 还可以用来考察搜索向推荐的导流做得好不好，如果成功把一部分来小红书做搜索的用户吸引到推荐页面，则比值会提升。

2.1.2 用户留存

用户留存率（retention rate）衡量 APP 留住用户的能力。有两种留存指标，一个种做“次 n 日内留存率（次 n 留）”，另一种叫做“第 n 日留存率（第 n 留）”。次 n 留的意思是今天（$t+0$）使用 APP 的用户中，有多大比例的用户在未来 $t+n$ 天内使用至少一次 APP。第 n 留的意思是今天（$t+0$）使用 APP 的用户中，有多大比例的用户在第 $t+n$ 天使用 APP。工业界使用最多的指标为次 1 留和次 7 留。次 n 留指标会滞后 n 天，也就是说实验上线 n 天之后才能计算次 n 留。次 28 留通常不作为短期 AB 测试的观测指标，只作为长期观测指标。

例 2.1

2 月 1 日有 1 亿用户使用某 APP。这 1 亿用户中，有 6 千万人在 2 月 2 日使用了该 APP。2 月 1 日的次 1 留、第 1 留都等于 60%。请注意，当 $n=1$ 时，两种留存指标相等。

♣

例 2.2

2 月 1 日有 1 亿用户使用某 APP。这 1 亿用户中，有 8 千万人在 2 月 2 日到 8 日之间使用该 APP 至少一次，那么 2 月 1 日的次 1 留等于 80%。这 1 亿用户中，有 3 千万人在 2 月 8 日当天使用该 APP 至少一次，那么 2 月 1 日的第 7 留等于 30%。请注意，次 n 留总是大于等于第 n 留，且次 n 留随 n 单调递增。通常来说，第 n 留随 n 增加而减小。

♣

用户生命周期（lift time，LT）指标为一定时间窗口内，用户使用 APP 天数的期望。设第 n 日留存为 r_n，定义 LT_n 为

$$\mathrm{LT}_n \ = \ 1+\sum_{i=1}^{n} r_i.$$

上式中的 1 为今天（第 0 天）的留存率 $r_0=1$。不难发现，LT_n 的值介于 1 和 $n+1$ 之间。LT_7 和 LT_{28} 是常用的 LT 指标。

留存指标都有个缺陷：真正地让用户变得更活跃，或者让不活跃的用户不再登录，那么留存指标都会增长。做一个假想实验：把留存率低的用户的账户全部关闭，能登录的用户都比较活跃，留存指标会增长，出现虚假的繁荣。反之，如果有策略能让流失的用户变成低活用户，原本是好事，反倒会拉低留存指标。如果发现留存指标增长，而 DAU 下跌，那么就应当分析是不是策略赶走了不活跃的用户；如果拆分人群分析的话，可能会发现高活用户指标持平，而低活用户的 DAU 和留存下跌。

此外，能在短期内提升留存指标的实验非常少。我们从未见过靠优化体验的策略能提升留存指标、且不会损害消费指标。提升留存是推荐系统的最高目标，能提升留存、但是损害消费的策略是值得推全的。

2.1.3 消费

人均阅读量是指平均每天每位用户阅读了多少物品的详情页。对于小红书的双列推荐，曝光不算阅读，只有点击进入笔记才算阅读。对于小红书的视频内流，每次下滑都会看到一个视频的详情，算增加一次阅读。值得注意的是，像小红书这样的 APP 有很多推荐场景，比如双列曝光、视频内流、子频道，这些场景各自的人均阅读量是此消彼长的，比如视频内流阅读量的提升会造成双列曝光阅读量的下降。所以我们需要统计所有场景的指标的加和或加权和，作为推荐系统人均阅读量。

人均使用推荐时长是指平均每天每位用户花多长时间使用推荐系统。以小红书为例，刷双列曝光、阅读笔记、播放视频、阅读评论都计入使用推荐的时长。人均使用 APP 时长大于等于使用推荐的时长，因为用户进入 APP 会使用搜索、聊天等功能。人均使用 APP 时长受推荐、搜索等多种因素的影响，未必适合作为推荐的核心指标。

在推荐系统的实验中，常见留存持平，人均阅读量、人均使用推荐时长这两种指标一涨一跌。举个例子，如果长视频的推荐体验优化得更好，那么用户对长视频的兴趣会增加，使用推荐的时长会增加，但相应地会减少阅读笔记和播放视频的数量。需要由数据分析的专家分析出两种消费指标的公平兑换关系，比如在没有更好的模型和策略的情况下，仅仅靠调推荐系统参数，每增加 1 次人均阅读量，会减少 1.2 分钟的人均使用推荐时长。当新策略对实验指标的影响优于兑换关系时，才能允许实验推全。

2.1.4 非核心指标

以小红书的首页推荐页为例，笔记以双列的形式曝光给用户，用户会点击感兴趣的笔记。统计每天的总点击数和总曝光数，取两者的比值，就是当天的点击率。有时用户会“手滑”误点，应当排除这样的无效点击。在点击发生之后，在笔记详情页停留时间超过 τ 秒，或者发生任意一种交互（点赞、收藏、关注等），就算一次有效点击。很显然，有效点击率小于等于点击率。

在用户进入笔记详情页之后，可能会发生点赞、收藏、转发、关注、评论等交互行为。发生交互行为通常意味着用户对笔记感兴趣，是对推荐系统的正反馈。交互行为，尤其是正面的评论，还能有效激励作者的发布积极性，对平台是非常有利的。用户将笔记转发到微信等平台，可以给小红书吸引到外部流量。我们会统计每天发生的各种交互行为的次数，求它们的加权和，作为交互指标。交互指标不是推荐系统的核心指标，即算法工程师不会把提升交互指标作为自己的工作方向，但交互指标也是实验中需要重点关注的。交互指标与核心指标往往有正相关性，但有的策略会增长核心消费指标、损失交互指标。当核心消费指标增长、交互指标下跌时，需要参考两者的兑换关系，只有当新策略对指标的影响优于兑换关系时，才能允许实验推全。

2.2 A/B 测试

推荐系统算法工程师的日常工作就是改进模型和策略，目标是提升推荐系统的业务指标。所有对模型和策略的改进，都需要经过线上 AB 测试，用实验数据验证模型和策略是否有效。本节介绍 AB 测试的基本概念。

2.2.1 实验组和对照组

举个例子，召回团队实现一种图神经网络（graph neural network，GNN）召回通道，离线实验的结果看上去非常好，那么下一步就是做线上的小流量 AB 测试，看 GNN 召回通道对线上业务指标的影响。此外，GNN 的神经网络结构中有一些需要调的参数，比如设置层数 $\in \{1, 2, 3\}$，具体该用几层的 GNN，也需要线上的 AB 测试确定。

AB 测试对用户做随机分桶，比如分成 10 桶，每桶占 10% 的流量，取一桶作为对照组，一个或多个桶作为实验组，在上述例子中，有 3 个实验组，分别是用 1 层、2 层、3 层的 GNN。统计各组的业务指标，比如 DAU、次 1 日留存、用户使用推荐时长、曝光笔记数、点击率、点赞率。计算每个实验组与对照组的差（diff），得到 3 个 diff，反映出新的 GNN 召回通道对业务指标的影响。如果有一组实验结果显著正向，则可以扩大它的流量，进一步观测实验的收益。

值得注意的是分桶必须要足够均匀，保证各桶所有业务指标都持平。均匀分 10 个桶，每个桶中有 10% 的用户，并不能保证每个桶的 DAU、中低活用户数、留存、消费、曝光次数、点击次数、交互次数都相等（至少精确到万分之一）。假如两个桶的某些核心指标有万分之几的差异，那么 A/B 测试测出的 diff 是没有意义的。分桶的时候，应当尽量保证各活跃度、人群的分桶是均匀的，避免桶间指标不平。

2.2.2 分层实验

以小红书这样的信息流产品为例，同时有多个团队做算法、前端界面、信息流广告、运营策略方面的优化，线上需要同时做数十个、甚至上百个 AB 测试。假如我们把用户随机分成 10 个桶，每个桶有 10% 的流量，取 1 个桶做对照组，剩下 9 个桶作实验组，那么线上最多只能同时开 9 个 AB 测试，根本无法满足产品迭代的需求。

解决方案就是分层实验，同层互斥，不同层正交。把实验分成很多层，比如召回、粗排、精排、重排、前端界面、等等。GNN 召回通道属于召回层的实验，它的流量跟召回层的实验互斥，与其他层的实验正交。举个例子，召回层和粗排层各自独立对用户做分桶，把召回层的桶记作集合 $\mathcal{U}_1, \cdots, \mathcal{U}_{10}$，把粗排层的桶记作集合 $\mathcal{V}_1, \cdots, \mathcal{V}_{10}$。设系统中一共有 n 位用户，那么每个桶中有 $|\mathcal{U}_i| = |\mathcal{V}_j| = \frac{n}{10}$ 位用户。

- 同层互斥。同层的桶交集为空：$\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$，$\forall i \neq j$。同层的两个实验用不同的桶，那么两个实验就是严格隔离开的，两个召回实验不会同时作用在一位用户上。如果 GNN 召回的实验占了 4 个桶，那么其他召回实验只能用剩余的 6 个桶。同层互斥的目的是避免一个用户同时被两个召回的实验影响；假如两个实验相互干扰，结果会变得不可控。

- 不同层正交。不同层的桶是正交的：$|\mathcal{U}_i \cap \mathcal{V}_j| = \frac{n}{100}$，这是因为两层的分桶是独立随机进行的，一个召回桶和一个粗排桶有 $\frac{n}{100}$ 位用户重叠。如图 2.1 所示，如果召回桶 $\mathcal{U}_3$ 使用了 GNN 召回，而且涨了业务指标，那么 10 个粗排桶的指标也会上涨，且 10 个粗排桶受到影响的程度相同，因此粗排桶 $\mathcal{V}_i$ 与 $\mathcal{V}_j$ 的实验对比仍然是公平的。

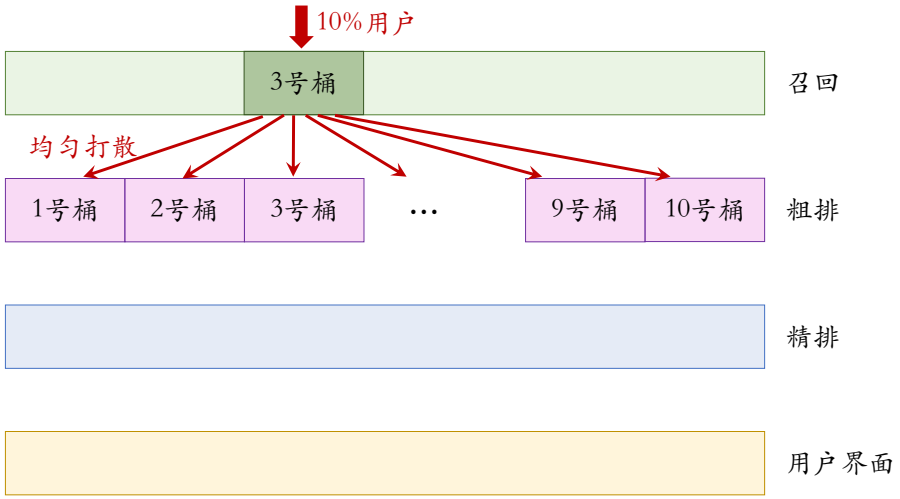

图 2.1: 召回层 3 号桶中有 10% 的用户，被均匀打散到粗排层的 10 个桶

大家思考一个问题：为什么允许召回和粗排实验同时作用到一位用户（正交），而不允许两个召回实验同时作用到一位用户（互斥）？如果两组实验同时作用到一位用户，实验效果可能会相互增强（$1+1>2$）或相互抵消（$1+1<2$）。两个召回实验相互增强或相互抵消的可能性很大，所以让召回实验互斥。而召回实验与粗排实验通常互不干扰（$1+1=2$），所以可以让召回和粗排正交。

例 2.3

我们添加召回通道 A 和召回通道 B，两者均会提升人均使用推荐时长 +1 分钟。但由于两条召回通道非常相似，假如同时添加，效果抵消，只会提升时长 +1 分钟，而不是 +2 分钟。

- 如果做互斥的实验，即两组实验的用户没有交集，则测量到两者的 diff 均为 +1 分钟，符合事实。
- 如果做正交实验，各用 50% 用户作实验组、50% 用户作对照组，那么如图 2.2 所示，实验 A 和 B 的对照组都“不干净”，比本该干净的对照组高了 $1 \times 0.25/0.5 = 0.5$ 分钟的时长，这导致实验组与对照组的 diff 等于 $1 - 0.5 = 0.5$ 分钟。测量结果为召回通道 A 和 B 各提升 +0.5 分钟，而实际上两者各提升 +1 分钟。

正确的方法是同层互斥，推全收益大的实验（比如召回通道 A），并在推全 A 的基础上做召回通道 B 的实验。

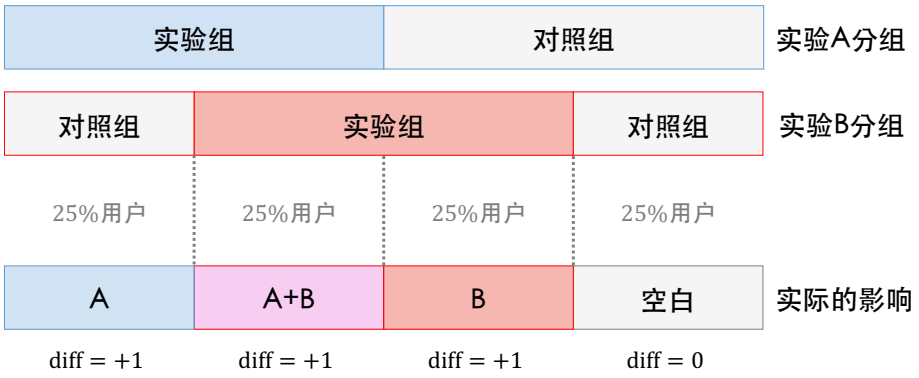

图 2.2: 如果召回通道 A 和 B 的效果抵消（1 + 1 = 1），那么测出两条召回通道的 diff 都等于 +0.5

2.2.3 holdout 机制

每个实验都有 AB 测试的指标收益，这些收益可以作为算法工程师个人的业绩。公司还需要对推荐部门进行绩效评估，比如每 2 个月考核一次部门整体对业务指标的贡献。不能简单地把所有推荐 AB 测试的指标收益相加作为部门的整体收益，在实践中，简单的加和与真正的叠加收益差距很大。

工业界常用 holdout 机制准确测算部门整体对业务指标的贡献。如图 2.3 所示，把推荐系统看做一个层，跟用户界面层、广告层正交。把推荐层内部划分成 10% vs 90% 互斥的两部分，其中 10% 用户的组作为 holdout 桶，90% 用户用作推荐实验。推荐层中 90% 的用户可以分成召回、粗排、精排、重排这样正交的层，每层都可以用全部 90% 的用户做实验。召回、粗排、精排、重排的有效实验均会带来跟 holdout 桶指标的 diff，且实验的 diff 会叠加（通常有折损）。如果公司以双月作为考核周期，那么每两个月结束的时候，计算 90% 实验流量和 10% holdout 流量各种指标的 diff（需要做归一化），作为整个推荐部门双月对业务指标的贡献。

在每个考核周期结束之后，会清除 holdout 桶，也就是让推全的实验从 90% 用户扩大到 100% 的用户。然后会把用户随机划分为 10% holdout 桶 vs 90% 实验桶，开始下一个考核周期。由于划分是随机的，新的 holdout 与实验桶的各种指标的 diff 都几乎为 0。随着召回、粗排、精排、重排的实验上线和推全，两个桶指标的 diff 会逐渐扩大。

2.2.4 实验推全与反转实验

所有的实验都是从小流量开始，如果业务指标的 diff 显著正向，则推全实验。举个例子，我们做一个召回通道的实验，取 1 个桶作为实验组，1 个桶作为对照组。如果观测到显著正向的业务指标收益，则推全这条召回通道，让它从 10% 的用户扩大到 90%。假设这条召回通道可以提升 +1 分钟的人均使用推荐时长。考察与 holdout 桶的 diff，那么在 10% 小流量实验期间，该召回通道贡献 $+\frac{1}{9}$ 分钟；在推全之后，它贡献了 +1 分钟。

上线一个有效的实验之后，需要观察很多指标，有的指标会立即被新策略影响，而有的指标有滞后性。点击率、点赞率、完播率等指标会立刻被新策略影响，实验上线 1

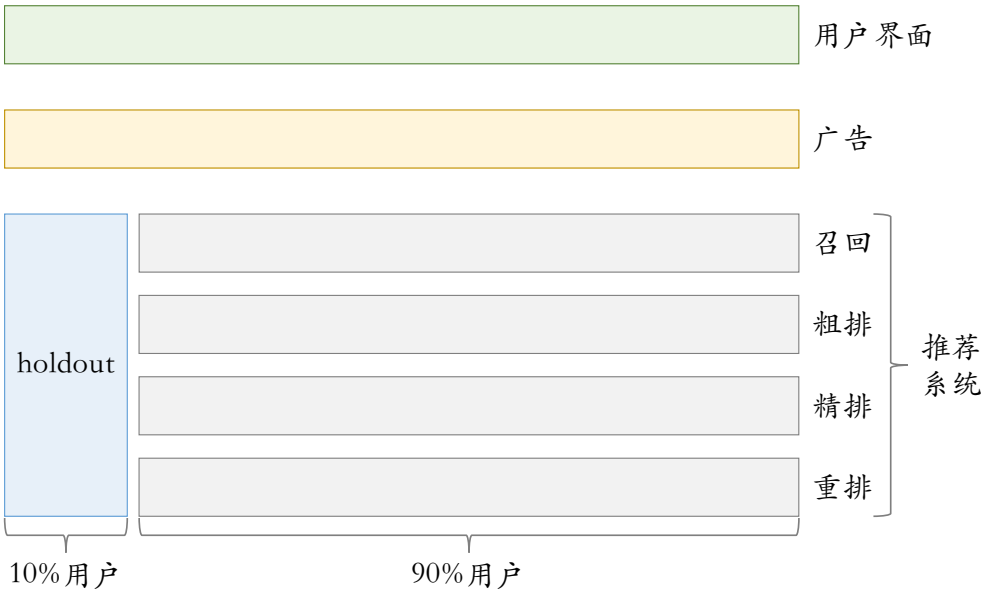

图 2.3: 推荐系统层中，holdout 桶占 10 的流量，实验桶用 90% 的流量。用户界面层与推荐系统层的流量正交，各自都有 100% 的流量。召回、粗排、精排、重排 4 个层正交，各自都有 90% 的流量

天、或者实验上线 10 天，观测到的指标不会差距太大。用户使用推荐的时长、曝光的物品数量这些指标有一点滞后，需要多观察几天，指标才能稳定。用户留存指标滞后非常严重，有可能短期内观测不到显著变化，但是在之后几个月中持续改善。指标滞后的原因不难理解，新策略改善用户体验，需要一些时间才能被用户感受到，感受到之后，用户对产品的粘性才越来越高。

算法工程师希望在小流量实验观测到显著收益之后，尽快推全新策略，这样可以腾出桶供其他实验使用，而且有时需要基于新策略做后续迭代优化。很多核心业务指标却具有滞后性，过快推全则观测不到完整的实验收益。实践中常用反转实验解决这对矛盾。如图 2.4 所示，每推全一个实验就新建一个推全层，它覆盖 90% 用户，新的推全层与召回等层正交。在推全层中，85% 的用户使用新策略，5% 的用户（反转桶）使用旧策略，这样就可以长期观测新策略与旧策略业务指标的 diff。当考察周期结束，清除 holdout 桶时，将新策略应用到 holdout 桶的 10% 用户。当反转实验完成时，关闭实验，则新策略会应用到反转桶的 5% 的用户。

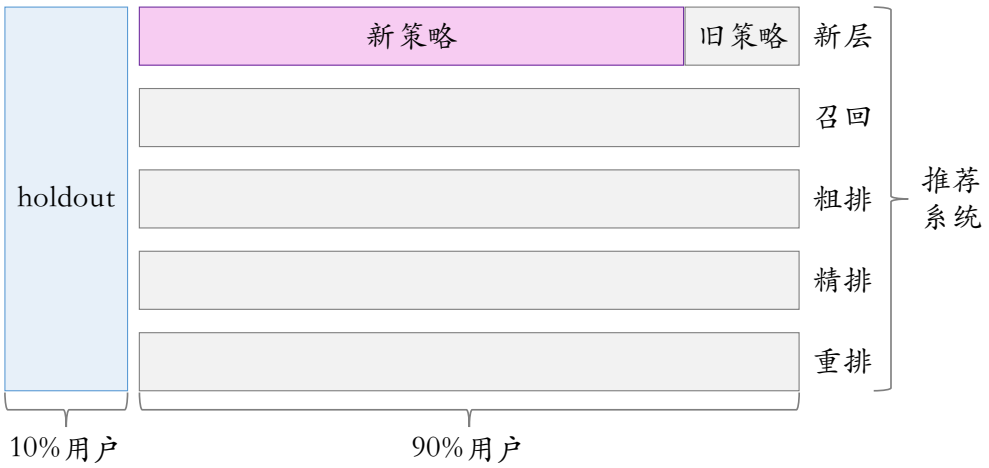

图 2.4: 在推全时，新建一层，包括 90% 的用户，它与推荐系统的其他层正交。在新层中，保留 5% 的用户作为反转桶使用旧策略，其余 85% 的用户使用推全的新策略