第一部分：推荐系统基本概念

1、推荐系统的基本概念

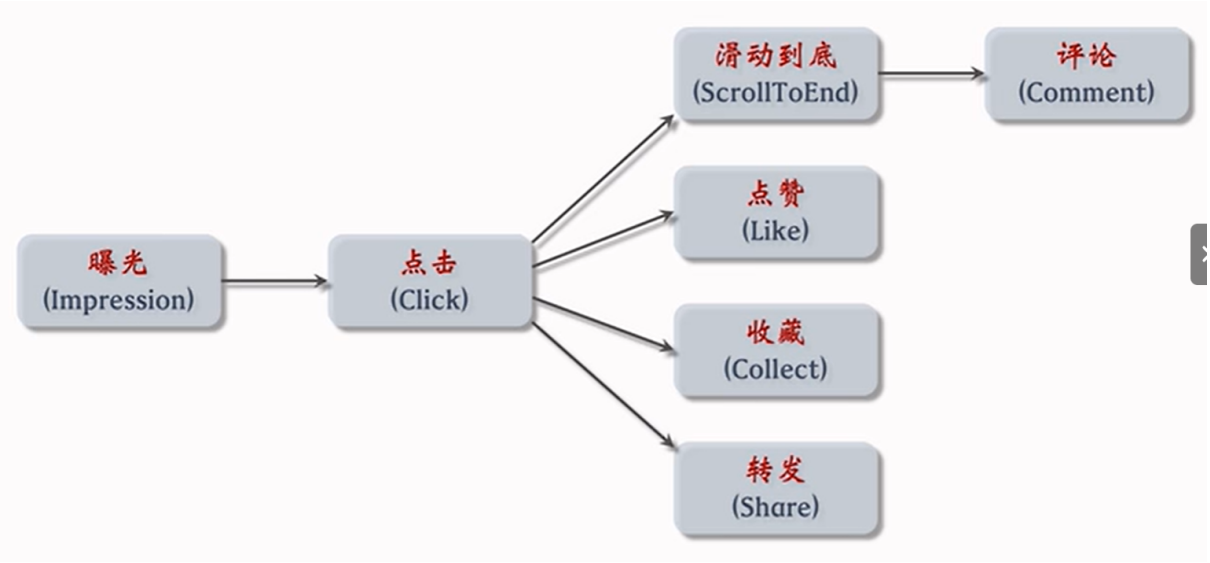
2、推荐系统的链路

3、推荐系统的线上实验

4、推荐系统的业务评估

1、推荐系统的基本概念

1.1 转化流程



不同公司、产品的转化流程可能不同

抖音没有曝光和点击

滑动到底（评论）、点赞、收藏、转发作为推荐系统的信号

1.2 消费指标（短期）

点击率=点击次数/曝光次数

点赞率=点赞次数/点击次数

收藏率=收藏次数/点击次数

转发率=转发次数/点击次数

阅读完成率=滑动到底次数/点击次数×f（归一化函数，与笔记长度有关）

1.3 北极星指标（最关键指标）

用户规模：日活用户数（DAU）、月活用户数（MAU）

消费：人均使用推荐时长、人均阅读笔记数量

发布：发布渗透率、人均发布量

*发布是小红书核心竞争力，其依赖于冷启动知识

北极星指标比点击率等消费指标更重要：通过提升内容多样性，虽然降低了点击率，但用户使用时间增长；反之则推送内容相同且单一，使用户减少新鲜感，用户活性降低，导致用户数量流失

1.4 实验流程

算法工程师：对模型特征、策略、系统进行改进，提升各种指标，提升推荐系统性能

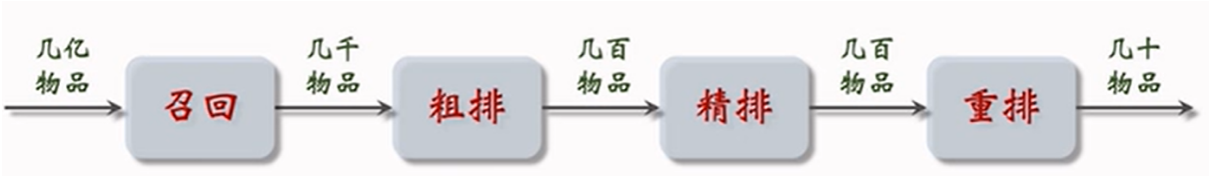
（1）离线实验：收集历史数据进行训练测试，未部署产品中，没有和用户交互，能一定程度判定系统好坏

（2）小流量A/B测试（线上实验）：把算法部署到实际产品中，用户实际跟算法交互（实验组用新策略，对照组用旧策略）

（3）全流量上线：如果新策略显著优于旧策略，可以加大流量最终推全

2、推荐系统的链路

以小红书为例：



更完整的步骤内容：

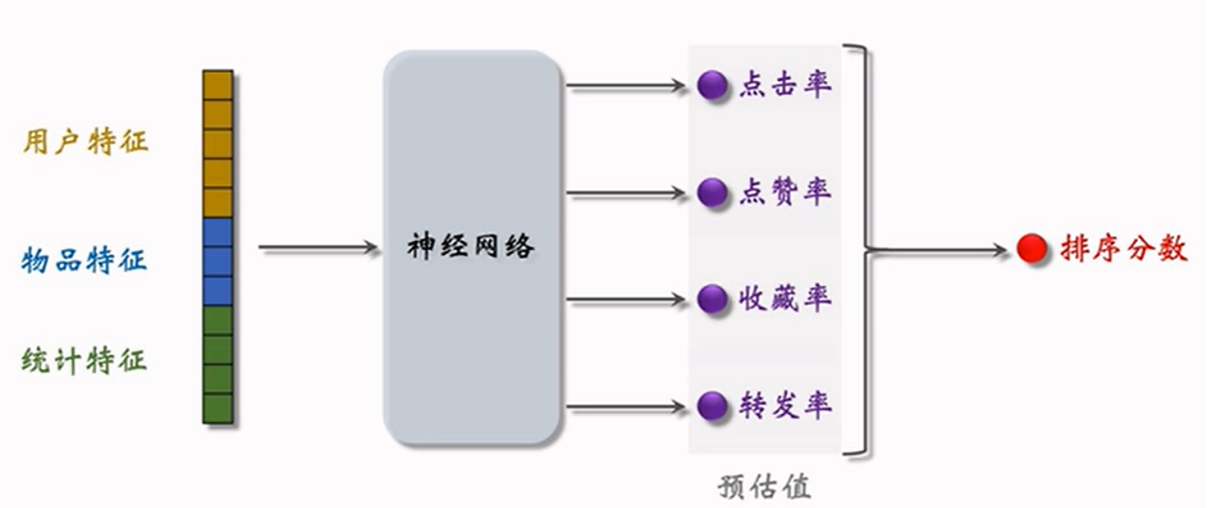
2.1 召回

从物品数据库中快速取回一些物品

（多召回通道，小红书每个通道取几十篇笔记，几十个通道一共就取出几千片笔记）

召回通道：协同过滤（排除掉用户不喜欢的作者、笔记、话题），双塔模型，关注的作者等

2.2 粗排和精排



粗排：用规模较小的机器学习模型对笔记打分，保留分数高的几百篇笔记（需要截断）

精排：用大规模神经网络对几百篇笔记进行打分，分数反映用户对笔记的兴趣（不用截断）

*粗排和精排的使用能很好平衡计算量和准确性

上图是对一篇笔记的打分，每个笔记有多个预估分数，最终融合成一个分数，作为给笔记排序的依据

粗排过程也会保证内容的多样性

2.3 重排

重排：根据精排分数和多样性分数做随机抽样得到几十篇笔记，然后打散加上广告内容

重排方式：做多样性抽样（MMR、DPP），从几百篇中选出几十篇，并用规则打散相似笔记

依据：精排分数、多样性

重排另一目的：插入广告、运营推广内容，根据生态要求调整排序（保证内容合适健康）

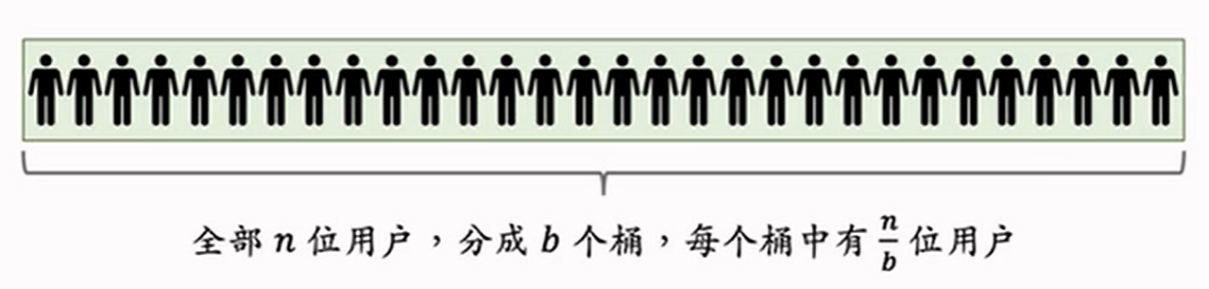
3、推荐系统的线上实验

3.1 A/B测试

目的：召回团队实现了一种GNN召回通道，离线实验结果正向，但离线实验指标提升并不代表线上实验也会有收益，因此要做线上小流量A/B测试，把新的召回通道给用户使用，观测用户真实行为数据（如日活、点击、交互等）

模型中的参数（如GNN的深度）需要用A/B测试选取最优参数

3.1.1 随机分桶（划分实验组和对照组）



用户数量足够大，每个桶的DAU、点击率等指标都相等

分桶方式：先用哈希函数把用户ID映射成某个区间内的整数，然后把这些整数均匀随机分成b个桶

假设1-3号桶都是实验组，但召回通道GNN参数不同（如GNN深度）；4号桶作为对照组，不用GNN

假如用户落在1号桶，则用1号桶的策略；若落在4号桶，则不用GNN召回

需要计算每个桶的业务指标，如DAU、人均使用推荐的时长、点击率等

如果某个实验组指标明显优于对照组，则说明对应策略有效，值得推全（把流量推到100%，给所有用户都采用这种GNN策略）

3.1.2 分层实验（解决流量不够用的问题）

如果把用户随机分成10组，1组做对照，9组做实验，那么只能同时做9组实验，满足不了需求

分层实验：推荐系统（召回、粗排、精排、重排），用户界面，广告

（如GNN召回通道属于召回层）

同层互斥：GNN实验占了召回层4个桶，其他召回实验只能用剩余6个桶

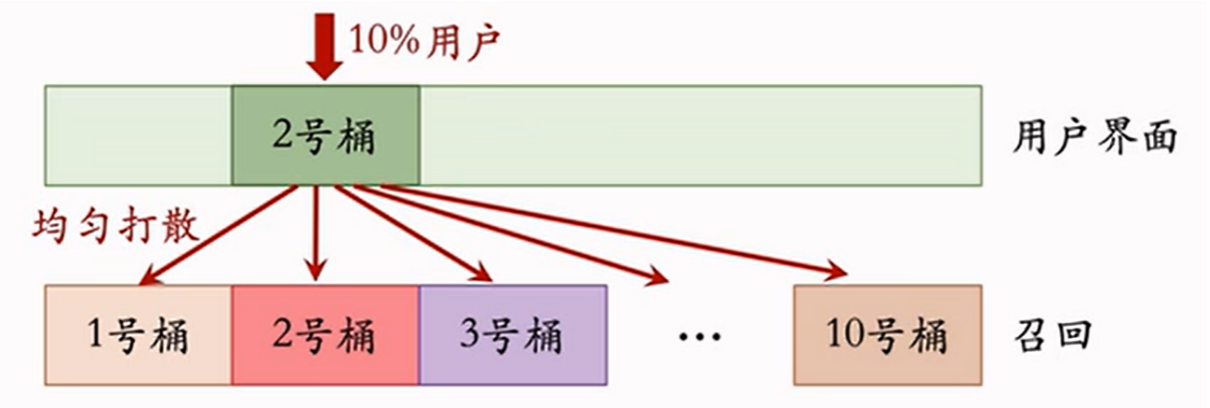
（避免同层用户同时被两个召回实验影响，否则相互干扰，实验不可控）

不同层正交：每一层独立随机对用户做分桶，每一层都可以独立用100%用户做实验

（不同层之间相互独立的随机思想）

举例说明：

- 召回层把用户分成 10 个桶：$\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_{10}$。
- 精排层把用户分成 10 个桶：$\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_{10}$。
- 设系统共有 n 个用户，那么 $|\mathcal{U}_i| = |\mathcal{V}_j| = n/10$。
- 召回桶 $\mathcal{U}_i$ 和召回桶 $\mathcal{U}_j$ 交集为 $\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$。
- 召回桶 $\mathcal{U}_i$ 和精排桶 $\mathcal{V}_j$ 交集的大小为 $|\mathcal{U}_i \cap \mathcal{V}_j| = n/100$。



3.1.3 互斥 vs 正交

（1）如果所有实验都正交，则可以同时做无数组实验

（2）同类策略（如精排模型中的两种结构）天然互斥，用户只能选其中一种

（3）同类策略（如添加两条召回通道）效果会相互增强（1+1>2）或相互抵消（1+1<2），互斥能避免同类策略相互干扰

*不同类型策略（如添加召回通道、优化粗排模型）通常不会相互干扰（1+1=2），可作为正交两层

4、推荐系统的业务评估

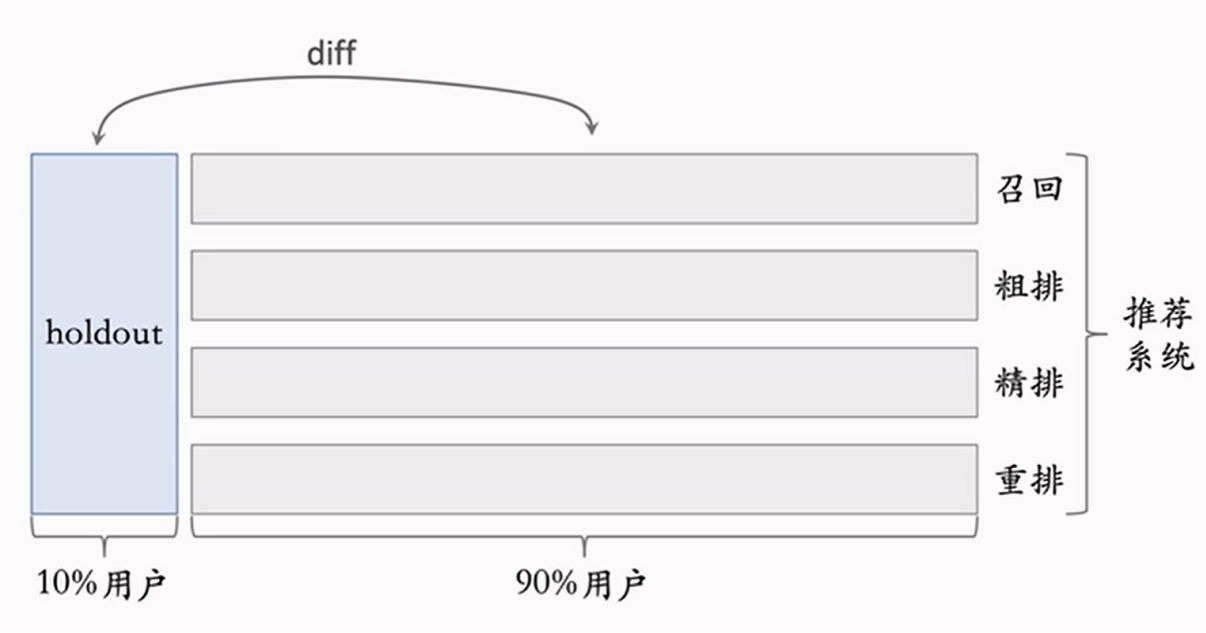
4.1 Holdout机制

用处：考察整个部门对业务指标的贡献

公司考察一个部门（推荐系统）在一段时间内对业务指标总体的提升，它不是每个实验（召回、粗排、精排、重排）单独对业务指标提升的总和，这样会有折损

策略：取10%用户作为holdout桶（相当于对照），推荐系统使用剩余90%用户进行实验，两者互斥

10%holdout桶vs90%实验桶的diff（需要归一化）即为整个部门业务指标收益



每个考核周期结束以后，清除holdout桶，让推全实验从90%用户扩大到100%用户

然后重新随机划分用户，得到holdout桶和实验桶，开始下一轮考核周期

由于随机均匀划分，初始阶段新的holdout桶与实验桶各指标diff接近于0，然后随着召回、粗排、精排、重排实验上线和推全，diff会逐渐扩大

4.2 实验推全

推荐实验从小流量开始。业务指标diff显示正向，则可关闭A/B线上实验，进行推全实验

推全实验在新层（与其他层正交）采用推全新策略，小流量10%用户的推荐系统指标提升，对应90%用户相应指标提升9倍。

4.3 反转实验

有的指标（点击、交互等）会立刻收到新策略影响；而有的指标（留存、推荐时长、人均阅读量等）初期变化不明显，有滞后性，要长期观测才能看到指标稳定

而算法工程师希望实验尽快观测到显著受益，进而尽快推全新策略。这样可以腾出其他桶给其他实验使用，或需要基于新策略做后续开发

这样的矛盾用**反转实验**解决：既保证新策略的推全，也可以长期观测实验指标

反转实验方式：推全的新层中开个旧策略的桶，用以长期观测实验指标

把反转桶保留很久，长期观察新策略和旧策略的diff

一个考核周期结束后，会清除holdout桶，清除holdout后会把新策略运用到holdout用户上，但不会影响反转桶（反转实验没有结束）

反转实验结束后，新策略会用到反转桶上，即实验真正推全，对100%用户有效