A/B测试

王树森

ShusenWang@xiaohongshu.com

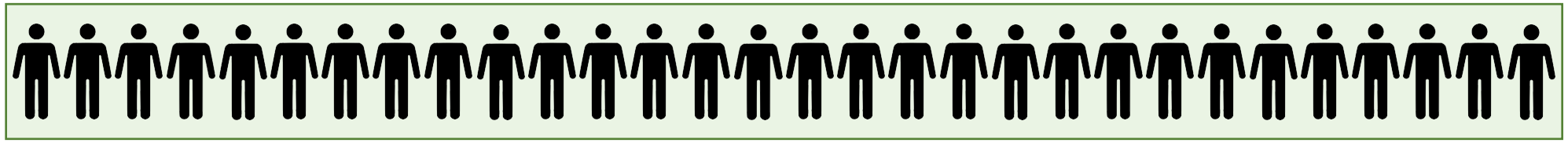
http://wangshusen.github.io/

A/B 测试

- 召回团队实现了一种 GNN 召回通道，离线实验结果正向。
- 下一步是做线上的小流量 A/B 测试，考察新的召回通道对线上指标的影响。
- 模型中有一些参数，比如 GNN 的深度取值 $\in \{1, 2, 3\}$，需要用 A/B 测试选取最优参数。

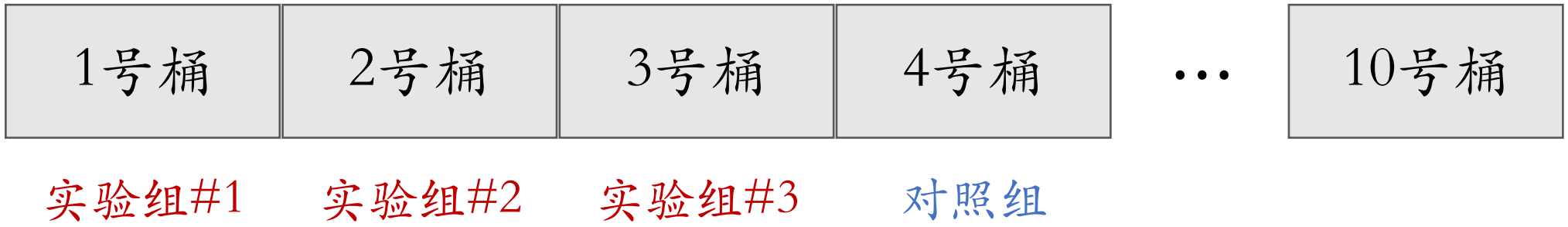
随机分桶

- 分 $b = 10$ 个桶，每个桶中有 10% 的用户。
- 首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。



全部 n 位用户，分成 b 个桶，每个桶中有 $\frac{n}{b}$ 位用户

随机分桶



- 计算每个桶的业务指标，比如 DAU、人均使用推荐的时长、点击率、等等。
- 如果某个实验组指标显著优于对照组，则说明对应的策略有效，值得推全。

分层实验

流量不够用怎么办？

- 信息流产品的公司有很多部门和团队，大家都需要做 A/B 测试。
  - 推荐系统（召回、粗排、精排、重排）
  - 用户界面
  - 广告
- 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。

分层实验

- 分层实验：召回、粗排、精排、重排、用户界面、广告……
  （例如 GNN 召回通道属于召回层。）
- 同层互斥：GNN 实验占了召回层的 4 个桶，其他召回实验只能用剩余的 6 个桶。
- 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% 的用户做实验。

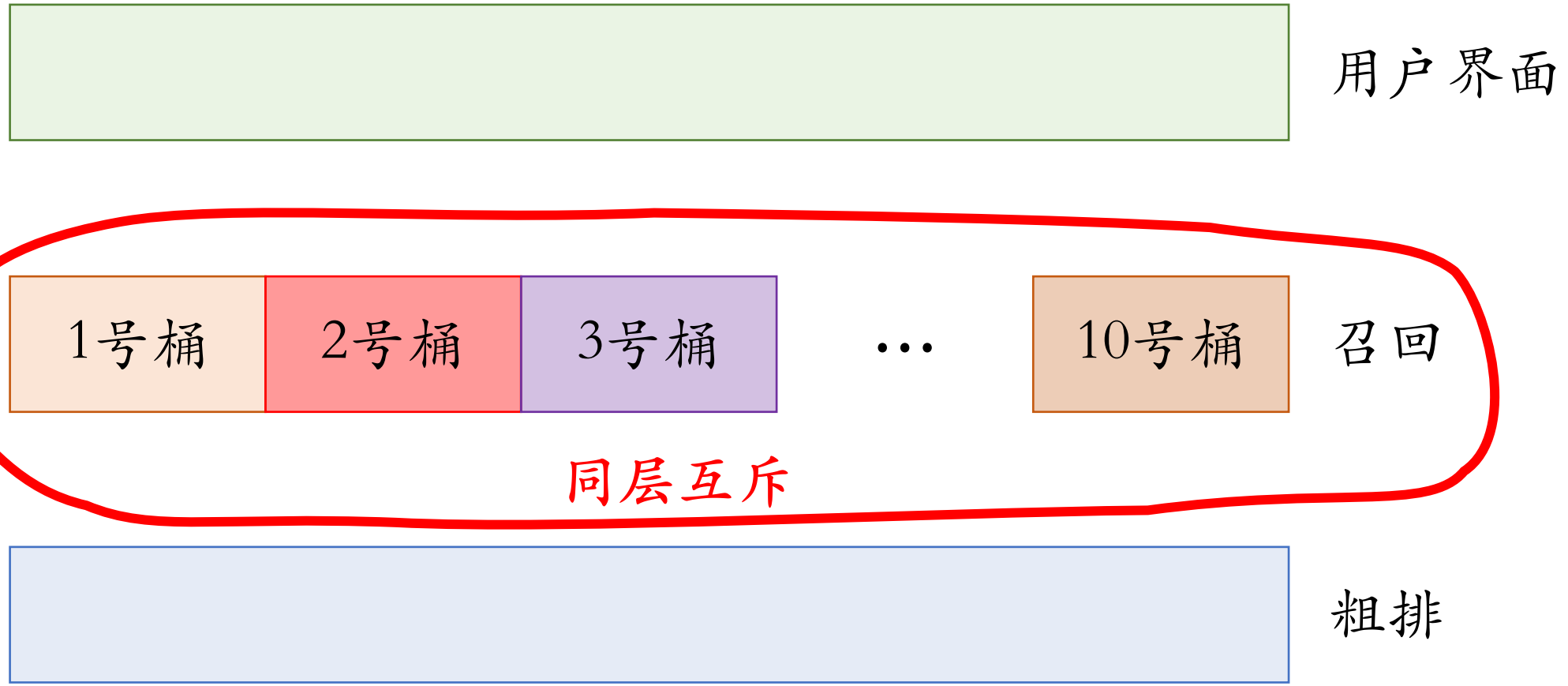
参考文献：

- Tang et al. Overlapping experiment infrastructure: more, better, faster experimentation. In *KDD*, 2010.

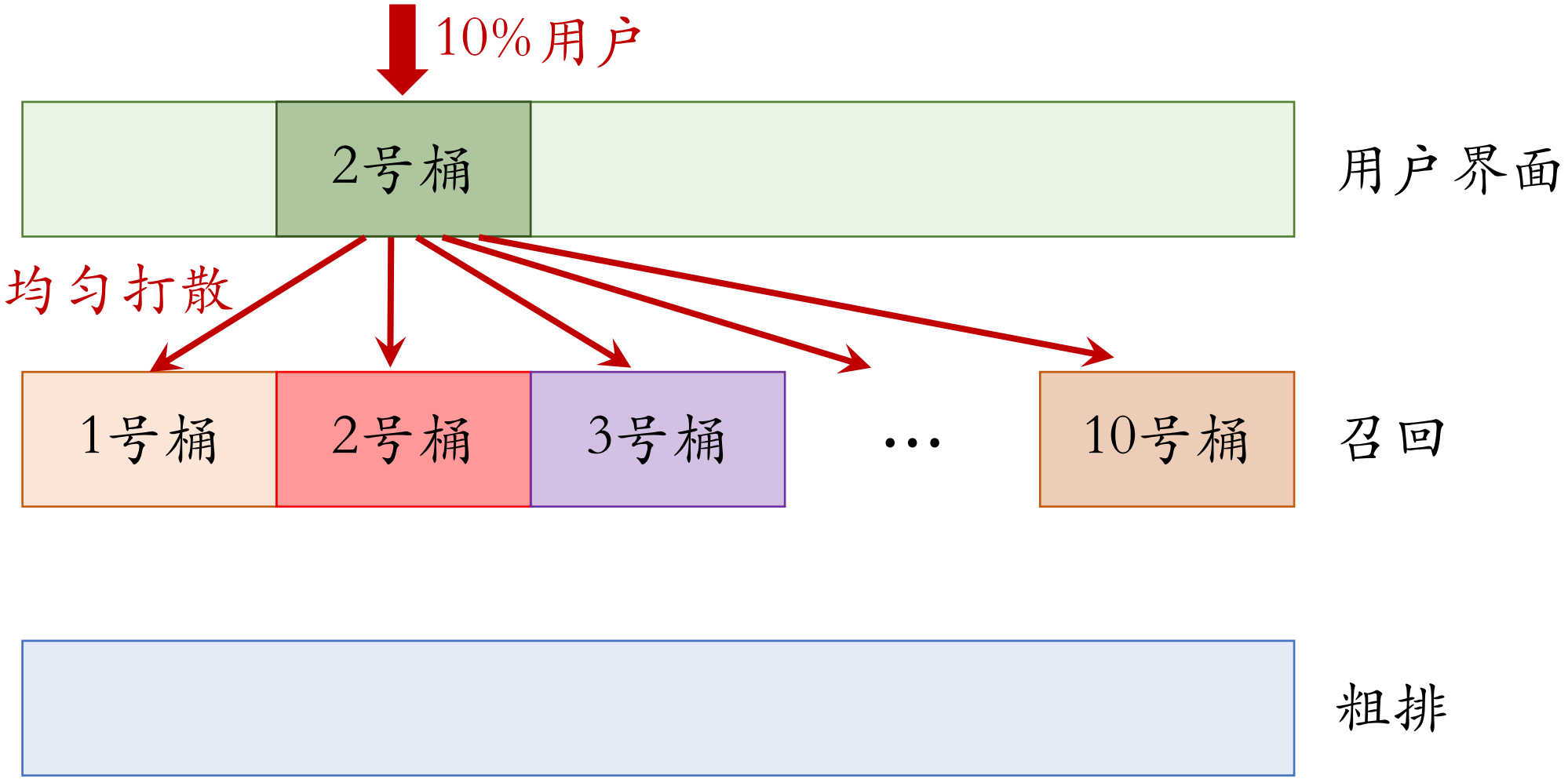
分层实验

- 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 $|\mathcal{U}_i \cap \mathcal{V}_j| = n/100$。

同层互斥



不同层正交



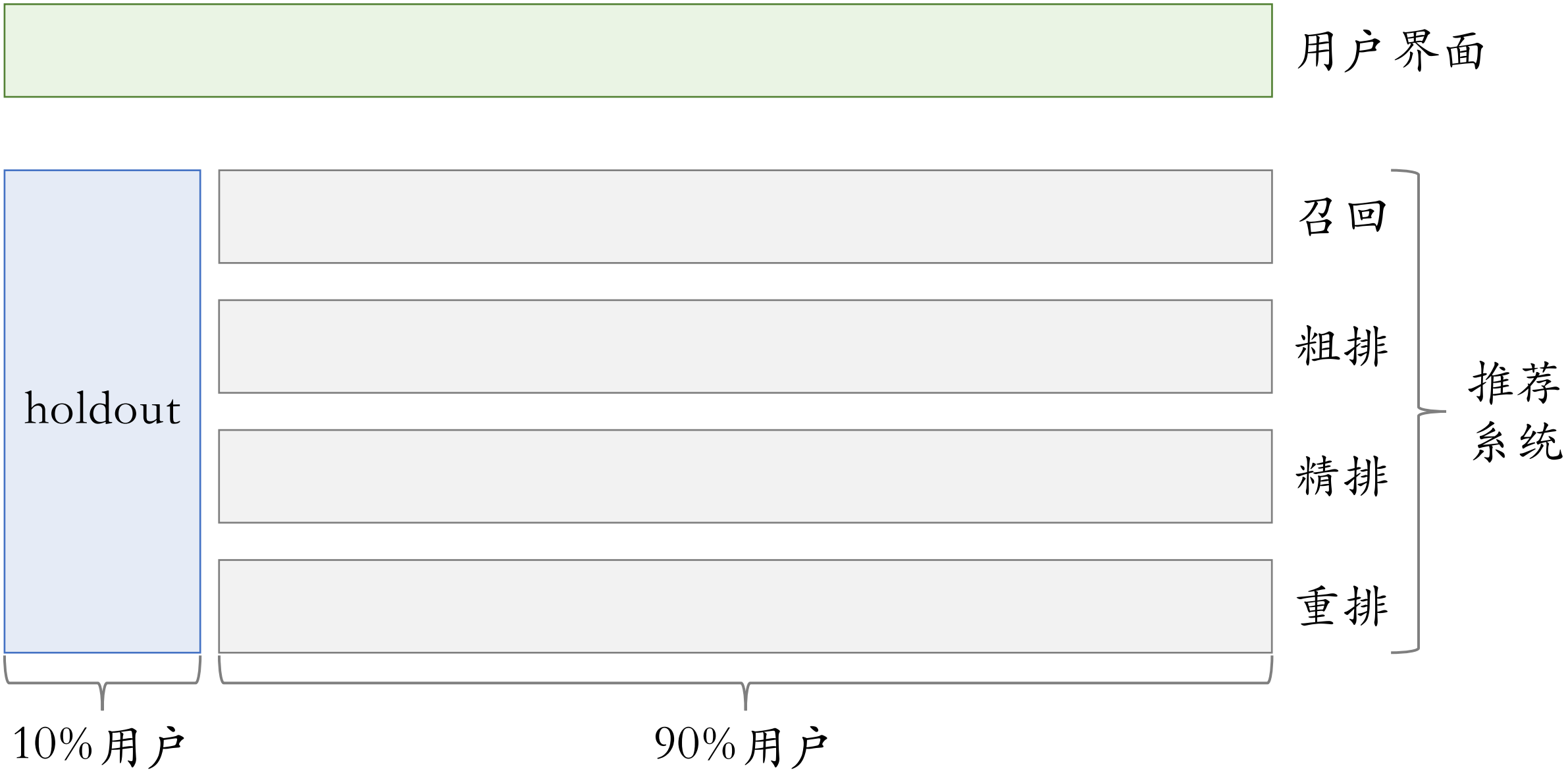
互斥 vs 正交

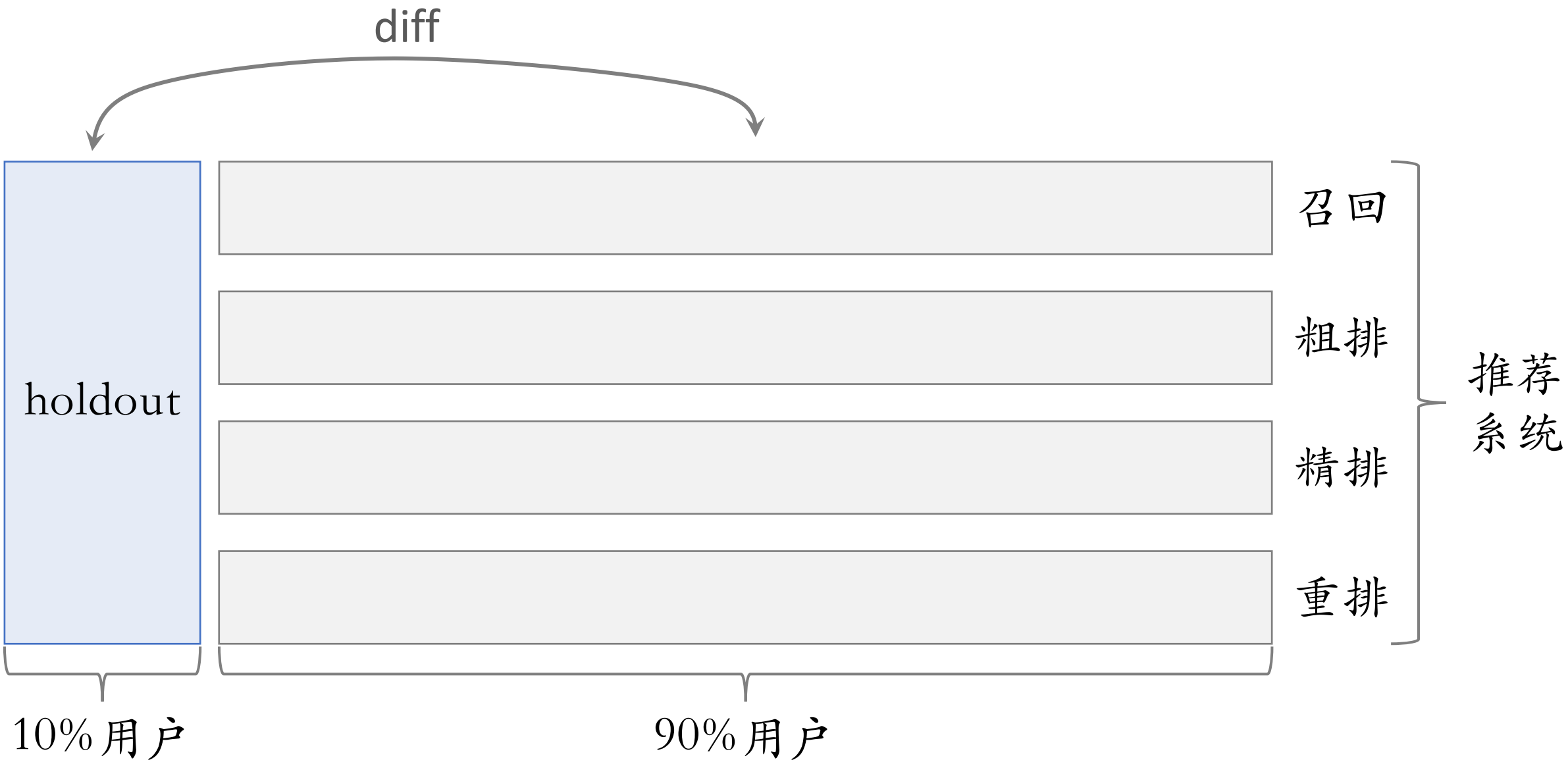
- 如果所有实验都正交，则可以同时做无数组实验。
- 同类的策略（例如精排模型的两种结构）天然互斥，对于一个用户，只能用其中一种。
- 同类的策略（例如添加两条召回通道）效果会相互增强（1+1>2）或相互抵消（1+1<2）。互斥可以避免同类策略相互干扰。
- 不同类型的策略（例如添加召回通道、优化粗排模型）通常不会相互干扰（1+1=2），可以作为正交的两层。

Holdout 机制

Holdout 机制

- 每个实验（召回、粗排、精排、重排）独立汇报对业务指标的提升。
- 公司考察一个部门（比如推荐系统）在一段时间内对业务指标总体的提升。
- 取 10% 的用户作为 holdout 桶，推荐系统使用剩余 90% 的用户做实验，两者互斥。
- 10% holdout 桶 vs 90% 实验桶的 diff（需要归一化）为整个部门的业务指标收益。

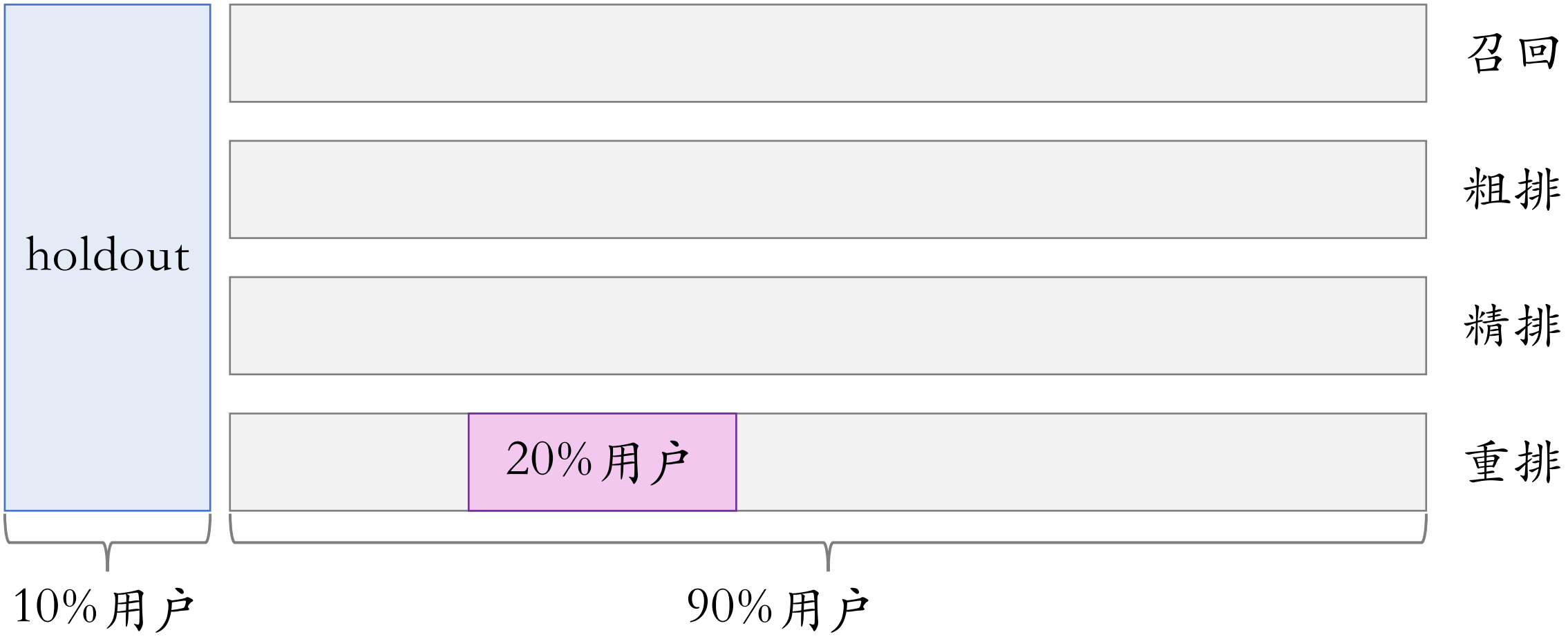

用户界面
holdout
召回
粗排
精排
重排
推荐系统
10%用户
90%用户


diff
holdout
召回
粗排
精排
重排
推荐系统
10%用户
90%用户

Holdout 机制

- 每个考核周期结束之后，清除 holdout 桶，让推全实验从 90% 用户扩大到 100% 用户 。
- 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。
- 新的 holdout 桶与实验桶各种业务指标的 diff 接近 0。
- 随着召回、粗排、精排、重排实验上线和推全，diff 会逐渐扩大。

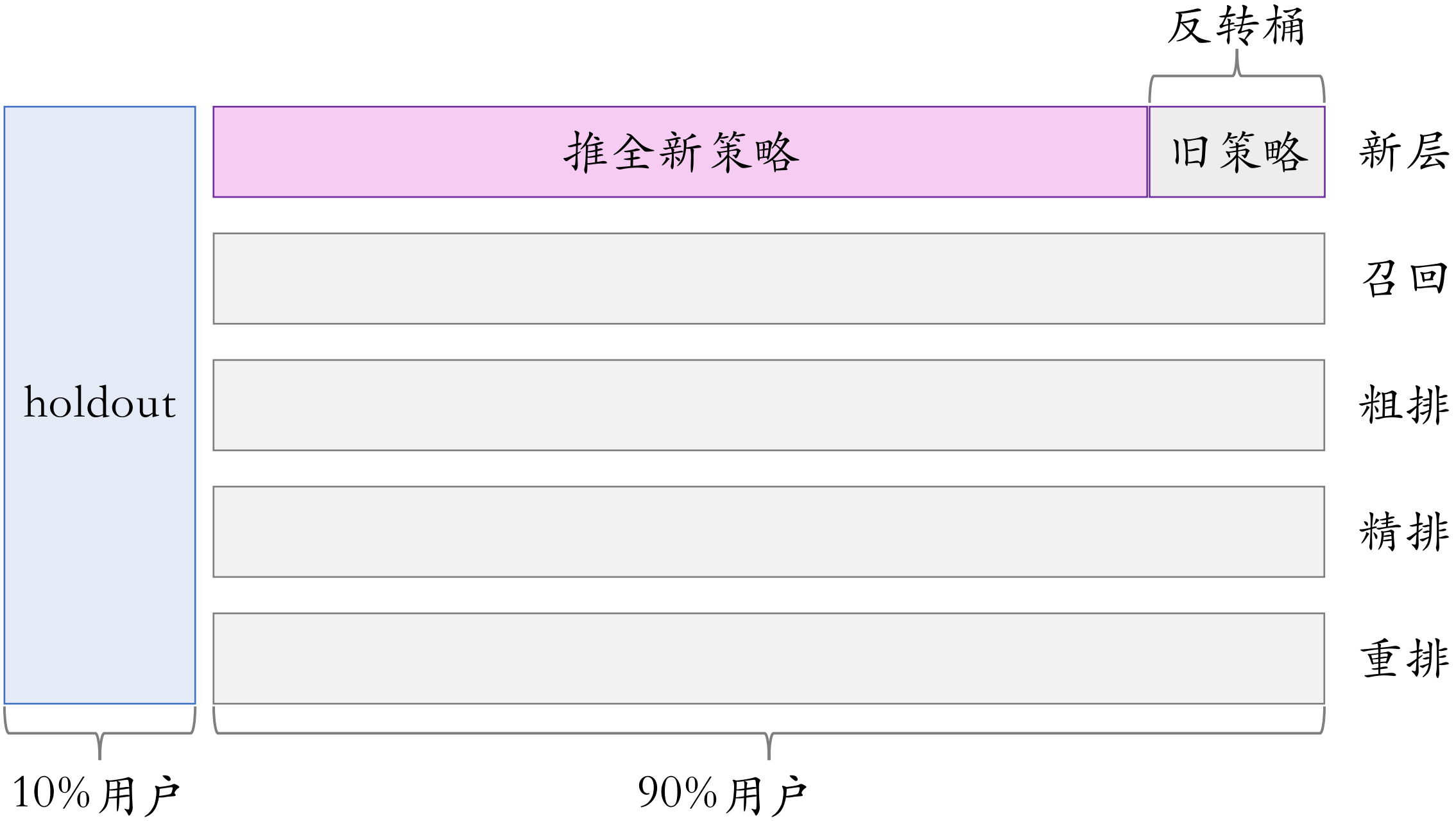
实验推全 & 反转实验


holdout
召回
粗排
精排
重排
20%用户
10%用户
90%用户


推全新策略
新层
召回
粗排
精排
重排
holdout
holdout
10%用户
90%用户

反转实验

- 有的指标（点击、交互）立刻收到新策略影响，有的指标（留存）有滞后性，需要长期观测。
- 实验观测到显著收益后尽快推全新策略。目的是腾出桶供其他实验使用，或需要基于新策略做后续的开发。
- 用反转实验解决上述矛盾，既可以尽快推全，也可以长期观测实验指标。
- 在推全的新层中开一个旧策略的桶，长期观测实验指标。


反转桶
推全新策略
旧策略
新层
召回
粗排
精排
重排
holdout
10%用户
90%用户

总结

- 分层实验：同层互斥（不允许两个实验同时影响一位用户）、不同层正交（实验有重叠的用户）。
- Holdout：保留 10% 的用户，完全不受实验影响，可以考察整个部门对业务指标的贡献。
- 实验推全：新建一个推全层，与其他层正交。
- 反转实验：在新的推全层上，保留一个小的反转桶，使用旧策略。长期观测新旧策略的 diff。

Thank You!

http://wangshusen.github.io/