涨指标的方法：排序模型

王树森

http://wangshusen.github.io/

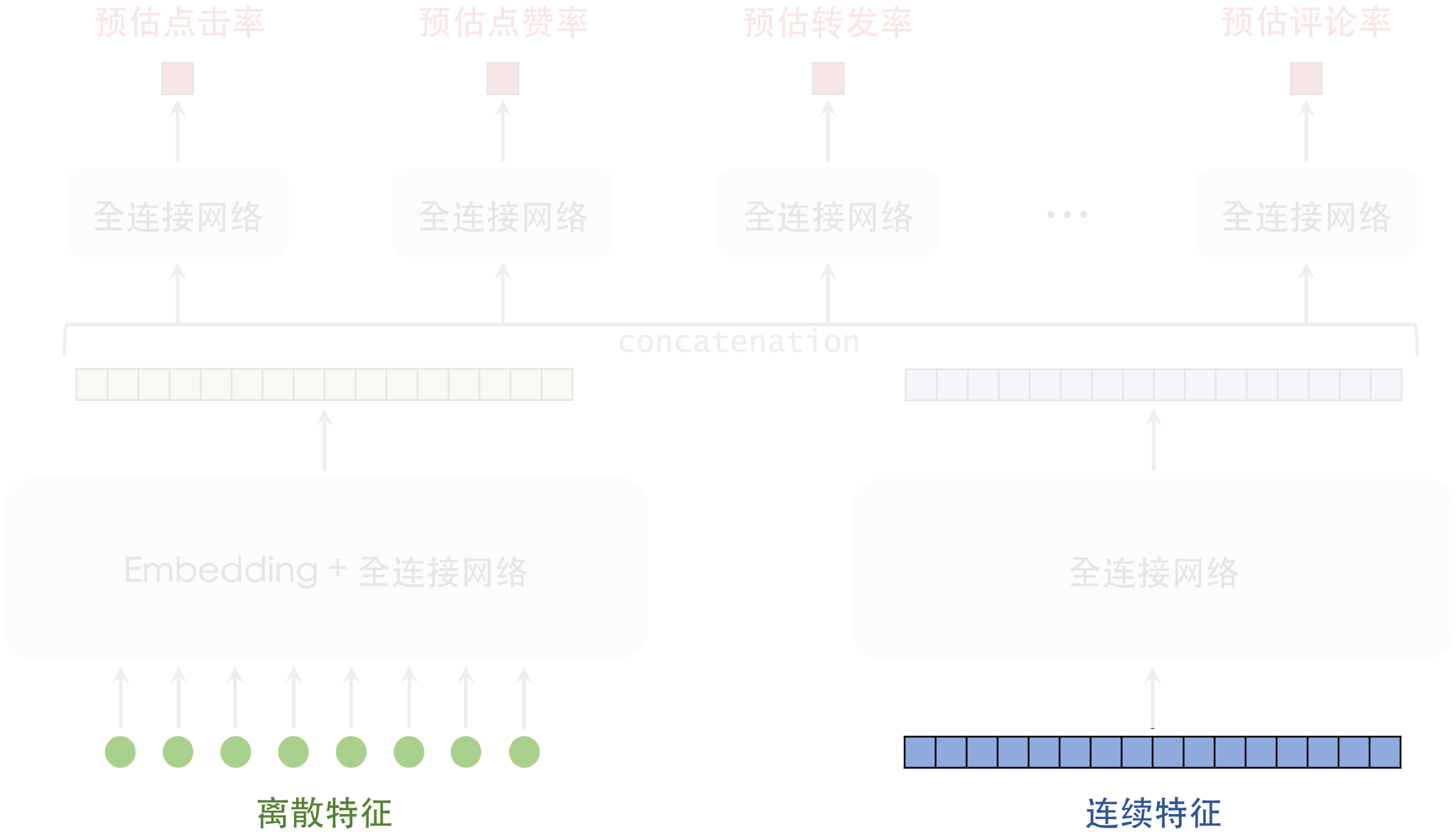
涨指标的方法有哪些？

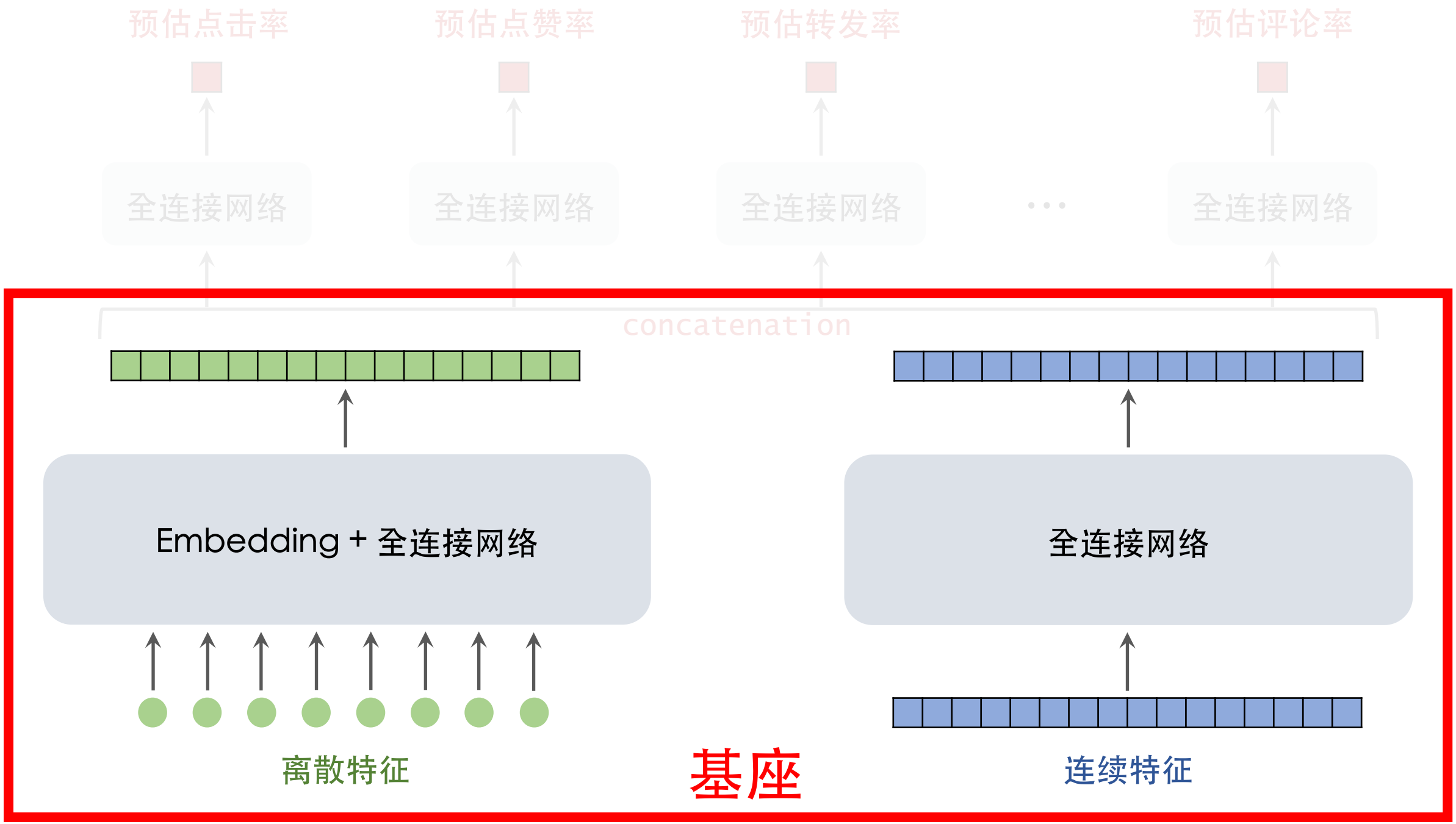
- 改进召回模型，添加新的召回模型。
- 改进粗排和精排模型。
- 提升召回、粗排、精排中的多样性。
- 特殊对待新用户、低活用户等特殊人群。
- 利用关注、转发、评论这三种交互行为。

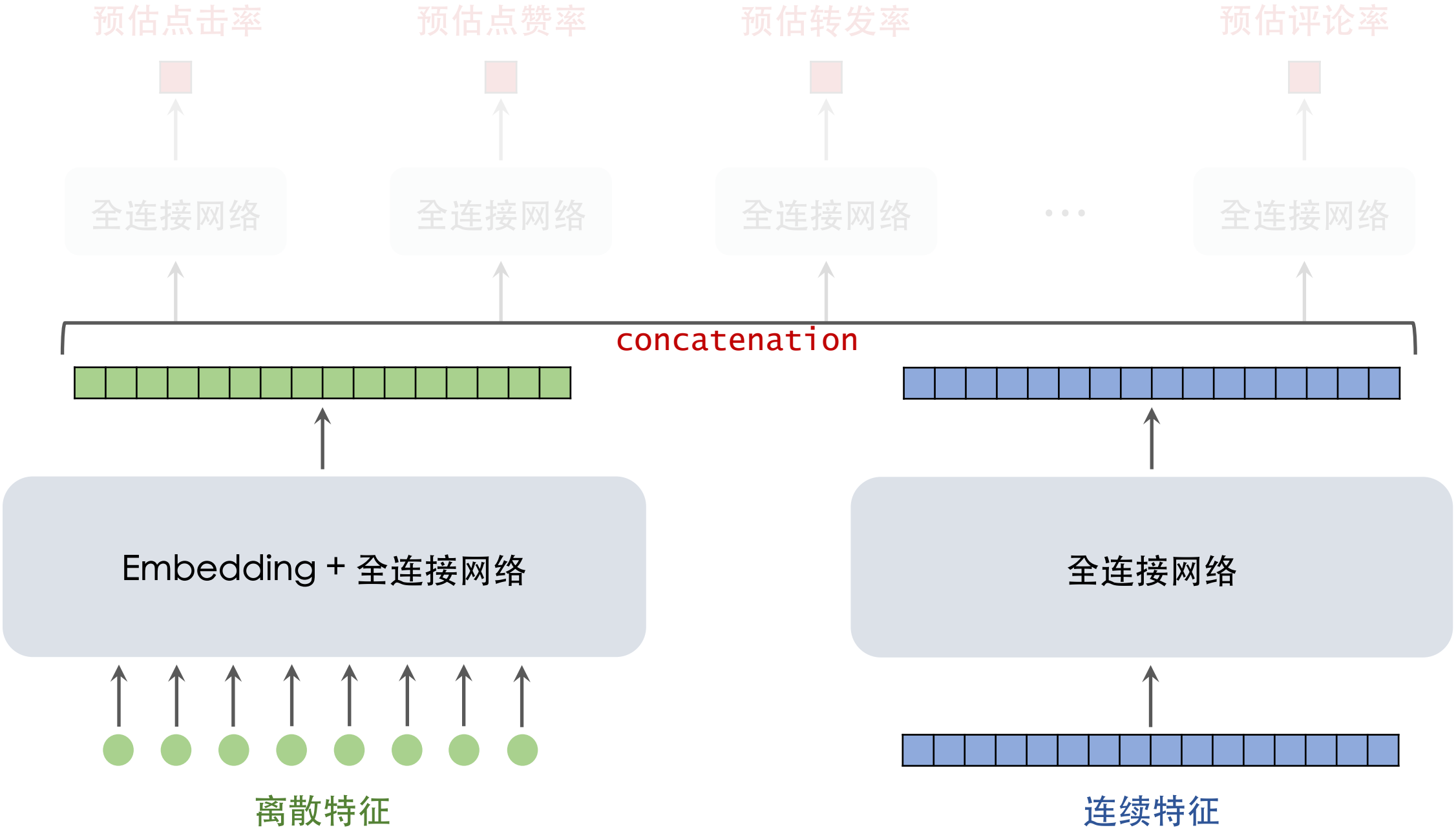
排序模型

1. 精排模型的改进
2. 粗排模型的改进
3. 用户行为序列建模
4. 在线学习
5. 老汤模型

精排模型的改进


预估点击率
预估点赞率
预估转发率
预估评论率
全连接网络
全连接网络
全连接网络
...
全连接网络
concatenation
Embedding + 全连接网络
全连接网络
离散特征
连续特征


预估点击率
预估点赞率
预估转发率
预估评论率
全连接网络
全连接网络
全连接网络
...
全连接网络
concatenation
Embedding + 全连接网络
全连接网络
离散特征
基座
连续特征


预估点击率
预估点赞率
预估转发率
预估评论率
全连接网络
全连接网络
全连接网络
...
全连接网络
concatenation
Embedding + 全连接网络
全连接网络
离散特征
连续特征

精排模型：基座

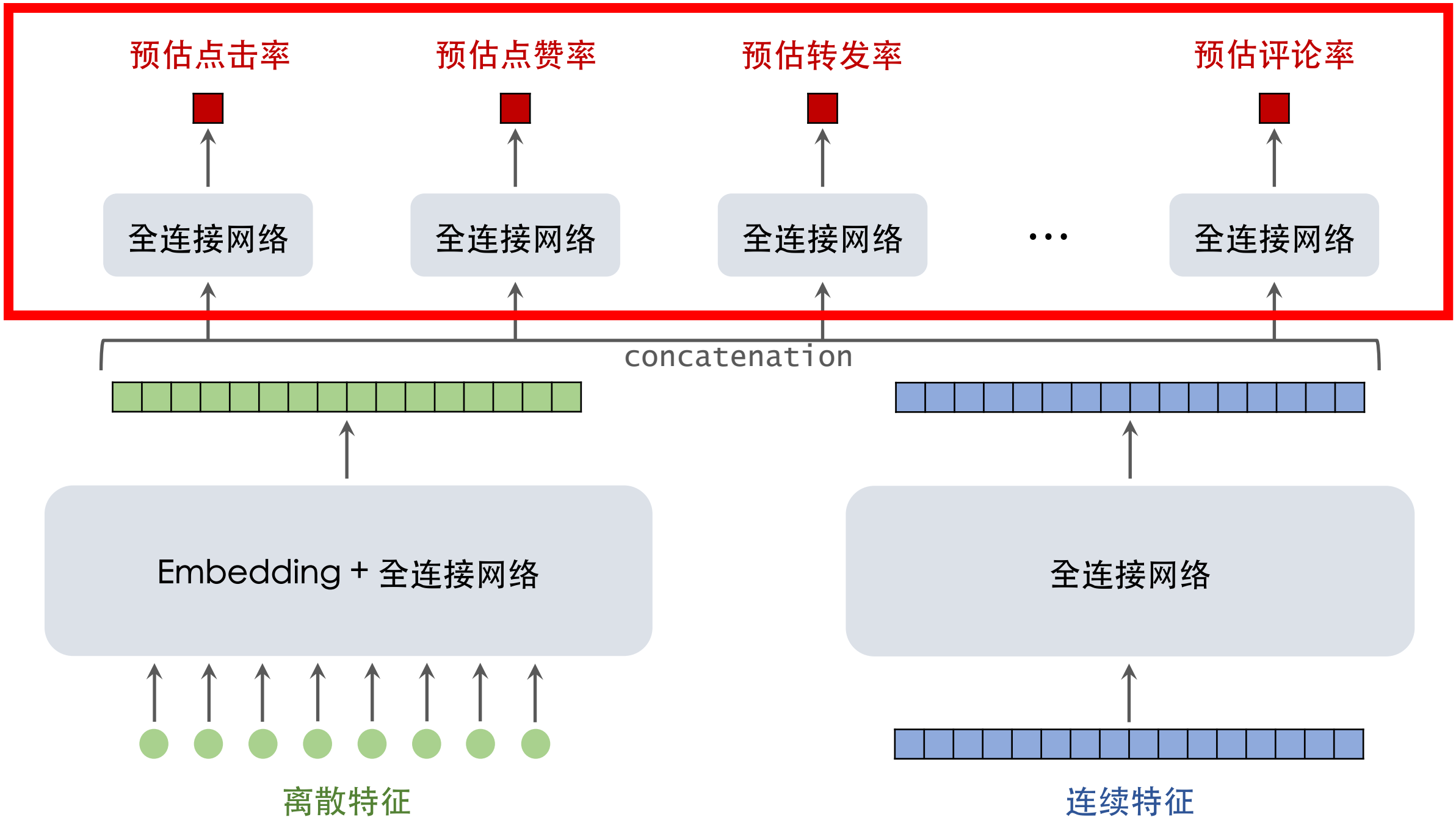
- 基座的输入包括离散特征和连续特征，输出一个向量，作为多目标预估的输入。
- 改进 1：基座加宽加深，计算量更大，预测更准确。

精排模型：基座

- 基座的输入包括离散特征和连续特征，输出一个向量，作为多目标预估的输入。
- 改进 1：基座加宽加深，计算量更大，预测更准确。
- 改进 2：做自动的特征交叉，比如 bilinear [1] 和 LHUC [2]。
- 改进 3：特征工程，比如添加统计特征、多模态内容特征。

参考文献

1. Huang et al. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *RecSys*, 2019.
2. Swietojanski et al. Learning hidden unit contributions for unsupervised acoustic model adaptation. In *WSDM*, 2016.


预估点击率
预估点赞率
预估转发率
预估评论率
全连接网络
全连接网络
全连接网络
...
全连接网络
concatenation
Embedding + 全连接网络
全连接网络
离散特征
连续特征

精排模型：多目标预估

- 基于基座输出的向量，同时预估点击率等多个目标。
- 改进 1：增加新的预估目标，并把预估结果加入融合公式。
  - 最标准的目标包括点击率、点赞率、收藏率、转发率、评论率、关注率、完播率……
  - 寻找更多目标，比如进入评论区、给他人写的评论点赞……
  - 把新的预估目标加入融合公式。

精排模型：多目标预估

- 基于基座输出的向量，同时预估点击率等多个目标。
- 改进 1：增加新的预估目标，并把预估结果加入融合公式。
- 改进 2：MMoE [1]、PLE [2] 等结构可能有效，但往往无效。
- 改进 3：纠正 position bias [3] 可能有效，也可能无效。

参考文献

1. Ma et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*, 2018.
2. Tang et al. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *RecSys*, 2020.
3. Zhou et al. Recommending what video to watch next: a multitask ranking system. In *RecSys*, 2019.

粗排模型的改进

粗排模型

- 粗排的打分量比精排大 10 倍，因此粗排模型必须够快。
- 简单模型：多向量双塔模型，同时预估点击率等多个目标。
- 复杂模型：三塔模型 [1] 效果好，但工程实现难度较大。

参考文献

1. Wang et al. COLD: towards the next generation of pre-ranking system. arXiv, 2020.

粗精排一致性建模

- 蒸馏精排训练粗排，让粗排与精排更一致。
- 方法1：pointwise 蒸馏。
  - 设 y 是用户真实行为，设 p 是精排的预估。
  - 用 $\frac{y+p}{2}$ 作为粗排拟合的目标。
  - 例：
    - 对于点击率目标，用户有点击（$y=1$），精排预估 $p=0.6$。
    - 用 $\frac{y+p}{2}=0.8$ 作为粗排拟合的点击率目标。

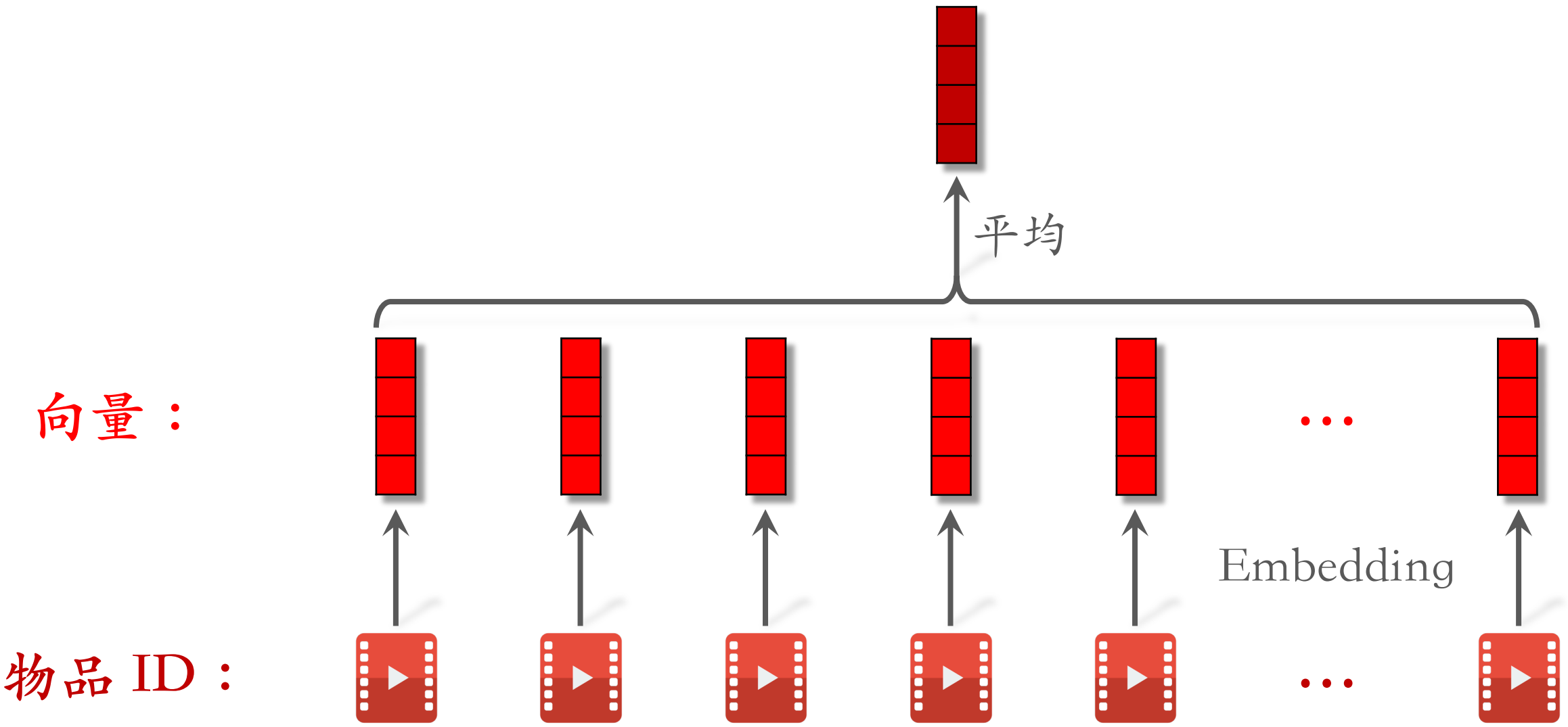
粗精排一致性建模

- 蒸馏精排训练粗排，让粗排与精排更一致。
- 方法1：pointwise 蒸馏。
- 方法2：pairwise 或 listwise 蒸馏。
  - 给定 k 个候选物品，按照精排预估做排序。
  - 做 learning to rank (LTR)，让粗排拟合物品的序（而非值）。
  - 例：
    - 对于物品 i 和 j，精排预估点击率为 $p_i > p_j$。
    - LTR 鼓励粗排预估点击率满足 $q_i > q_j$，否则有惩罚。
    - LTR 通常使用 pairwise logistic loss。

粗精排一致性建模

- 蒸馏精排训练粗排，让粗排与精排更一致。
- 方法1：pointwise 蒸馏。
- 方法2：pairwise 或 listwise 蒸馏。
- 优点：粗精排一致性建模可以提升核心指标。
- 缺点：如果精排出bug，精排预估值 p 有偏，会污染粗排训练数据。

用户行为序列建模


平均
向量：
…
Embedding
物品 ID：
…

用户行为序列建模

- 最简单的方法是对物品向量取平均，作为一种用户特征 [1]。
- DIN [2] 使用注意力机制，对物品向量做加权平均。
- 工业界目前沿着 SIM [3] 的方向发展。先用类目等属性筛选物品，然后用 DIN 对物品向量做加权平均。

参考文献

1. Covington, Adams, and Sargin. Deep neural networks for YouTube recommendations. In *RecSys*, 2016.
2. Zhou et al. Deep interest network for click-through rate prediction. In *KDD*, 2018.
3. Qi et al. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *CIKM*, 2020.

用户行为序列建模

- 改进1：增加序列长度，让预测更准确，但是会增加计算成本和推理时间。
- 改进2：筛选的方法，比如用类目、物品向量表征聚类。
  - 离线用多模态神经网络提取物品内容特征，将物品表征为向量。
  - 离线将物品向量聚为 1000 类，每个物品有一个聚类序号。
  - 线上排序时，用户行为序列中有 $n = 1{,}000{,}000$ 个物品。某候选物品的聚类序号是 70，对 n 个物品做筛选，只保留聚类序号为 70 的物品。n 个物品中只有数千个被保留下来。
  - 同时有好几种筛选方法，取筛选结果的并集。

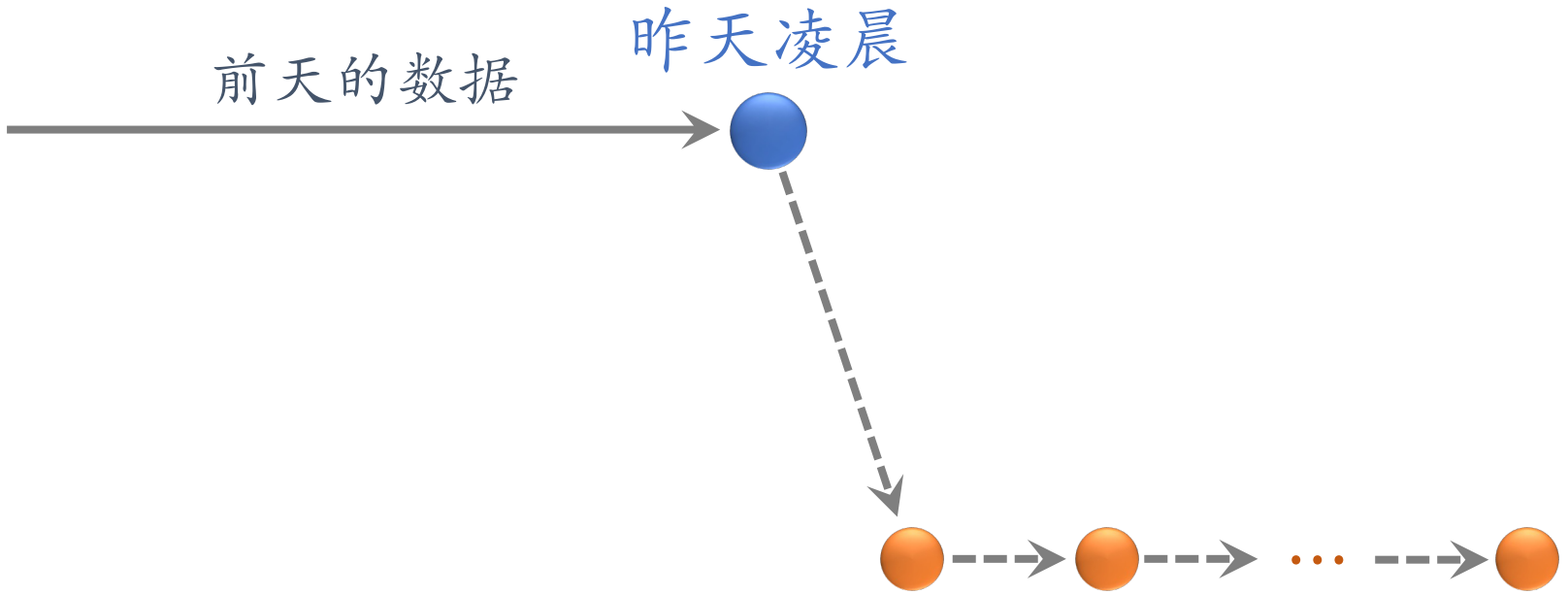
用户行为序列建模

- 改进1：增加序列长度，让预测更准确，但是会增加计算成本和推理时间。
- 改进2：筛选的方法，比如用类目、物品向量表征聚类。
- 改进3：对用户行为序列中的物品，使用 ID 以外的一些特征。
- 概括：沿着 SIM 的方向发展，让原始的序列尽量长，然后做筛选降低序列长度，最后将筛选结果输入 DIN。

在线学习

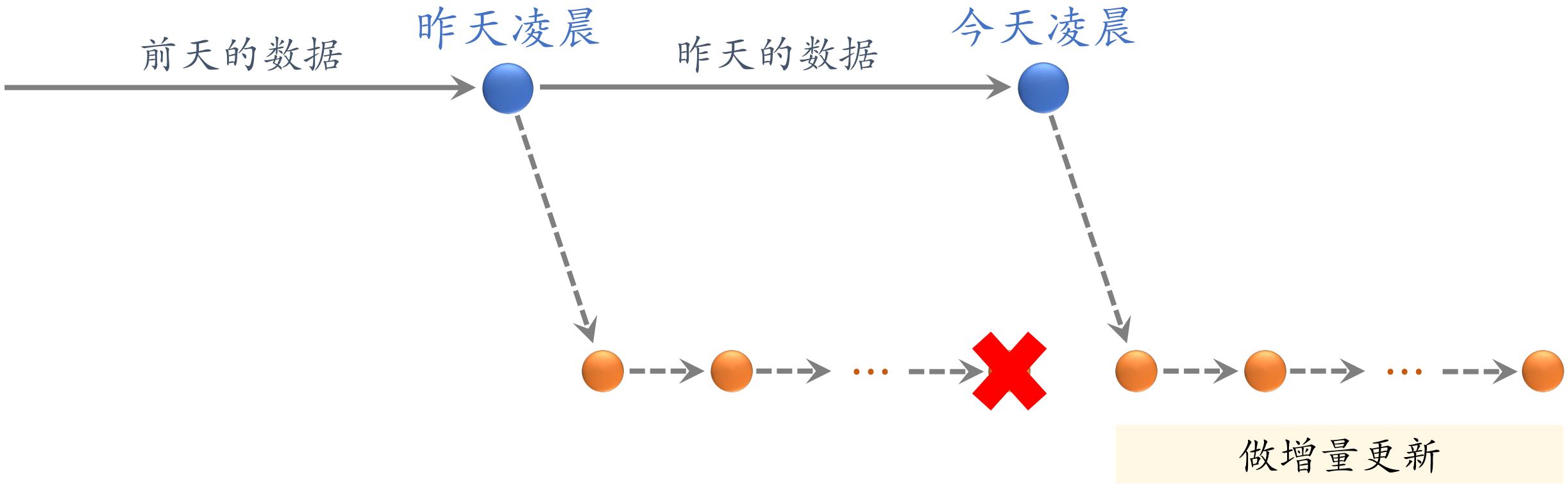
全量更新 vs 增量更新

基于前天的全量模型，用前天的数据，做全量更新。



全量更新 vs 增量更新

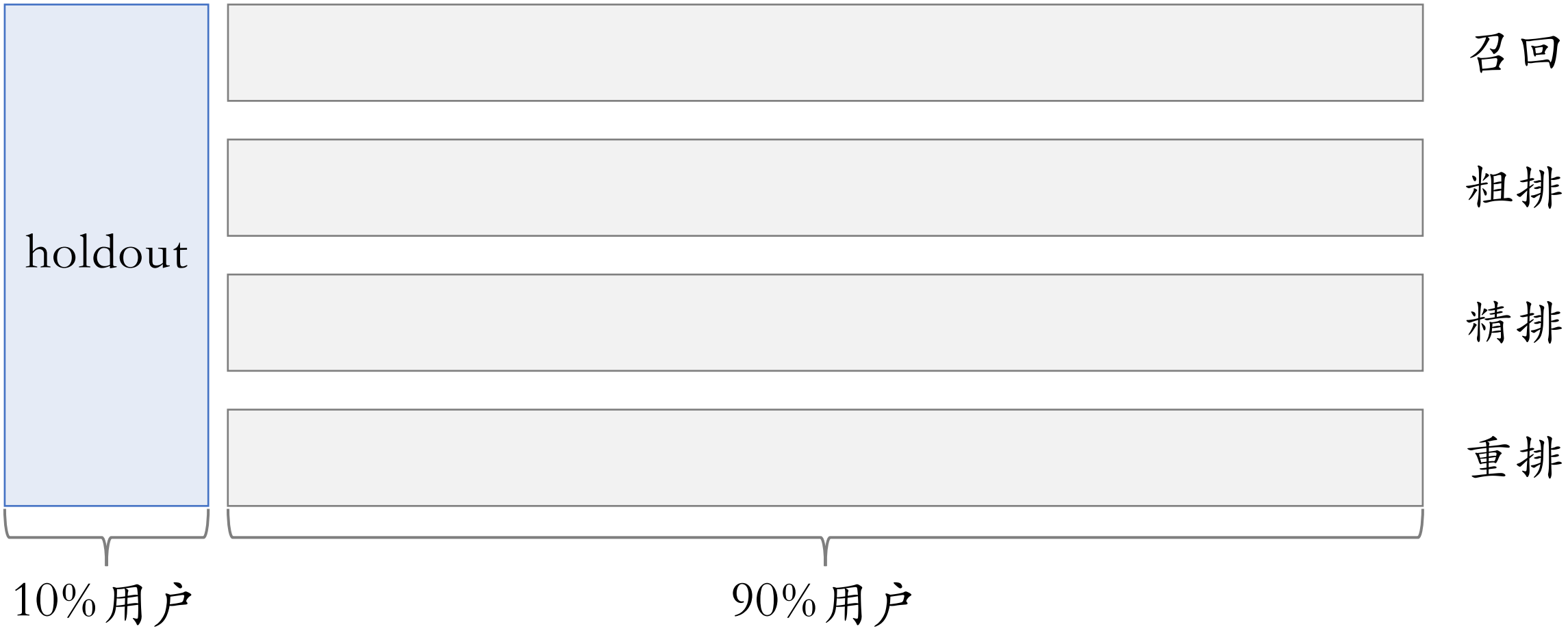
基于昨天的全量模型，用昨天的数据，做全量更新。



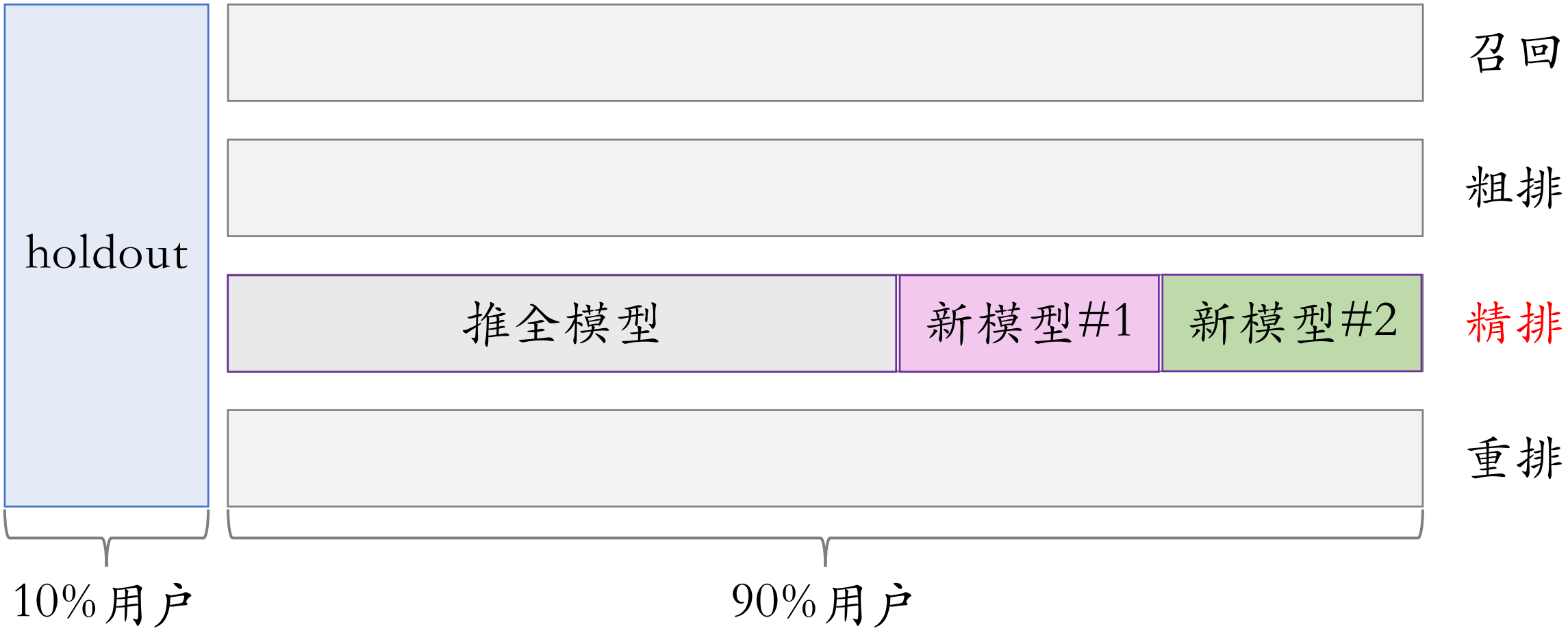
在线学习的资源消耗

- 既需要在凌晨做全量更新，也需要全天不间断做增量更新。
- 设在线学习需要 10,000 CPU core 的算力增量更新一个精排模型。推荐系统一共需要多少额外的算力给在线学习？
- 为了做 AB 测试，线上同时运行多个不同的模型。
- 如果线上有 m 个模型，则需要 m 套在线学习的机器。
- 线上有 m 个模型，其中 1 个是 holdout，1 个是推全的模型，$m-2$ 个测试的新模型。

在线学习的资源消耗



在线学习的资源消耗



在线学习的资源消耗

- 线上有 m 个模型，其中 1 个是 holdout，1 个是推全的模型，$m-2$ 个测试的新模型。
- 每套在线学习的机器成本都很大，因此 m 数量很小，制约模型开发迭代的效率。
- 在线学习对指标的提升巨大，但是会制约模型开发迭代的效率。

老汤模型

老汤模型

- 用每天新产生的数据对模型做 1 epoch 的训练。
- 久而久之，老模型训练得非常好，很难被超过。
- 对模型做改进，重新训练，很难追上老模型……
- 问题 1：如何快速判断新模型结构是否优于老模型？（不需要追上线上的老模型，只需要判断新老模型谁的结构更优。）
- 问题 2：如何更快追平、超过线上的老模型？（只有几十天的数据，新模型就能追上训练上百天的老模型。）

老汤模型

问题 1：如何快速判断新模型结构是否优于老模型？

- 对于新、老模型结构，都随机初始化模型全连接层。
- Embedding 层可以是随机初始化，也可以是复用老模型训练好的参数。
- 用 n 天的数据训练新老模型。（从旧到新，训练 1 epoch）
- 如果新模型显著优于老模型，新模型很可能更优。
- 只是比较新老模型结构谁更好，而非真正追平老模型。

老汤模型

问题 2：如何更快追平线上的老模型？

- 已经得出初步结论，认为新模型很可能优于老模型。用几十天的数据训练新模型，早日追平老模型。
- 方法 1：尽可能多地复用老模型训练好的 embedding 层，避免随机初始化 embedding 层。（Embedding 层是对用户、物品特点的“记忆”，比全连接层学得慢。）
- 方法 2：用老模型做 teacher，蒸馏新模型。（用户真实行为是 y，老模型的预测是 p，用 $\frac{y+p}{2}$ 作为训练新模型的目标。）

总结：改进排序模型

- 精排模型：改进模型基座（加宽加深、特征交叉、特征工程），改进多目标预估（增加新目标、MMoE、position bias）。
- 粗排模型：三塔模型（取代多向量双塔模型），粗精排一致性建模。
- 用户行为序列建模：沿着 SIM 的方向迭代升级，加长序列长度，改进筛选物品的方法。
- 在线学习：对指标提升大，但是会降低模型迭代升级效率。
- 老汤模型制约模型迭代升级效率，需要特殊技巧。

Thank You!

http://wangshusen.github.io/