WSS推荐系统学习笔记

Welcome to **WSS推荐系统学习笔记**。本笔记是基于王森树老师的推荐系统公开课进行整理，用于入门性的介绍推广搜算法，以应对接下来可能的推广搜实习笔试，面试以及工作。本学习笔记系统系的介绍了以下内容：

1. **概要** (概要 Quick Link)
   - 推荐系统的基本概念
   - 推荐系统的链路
   - AB测试
2. **召回** (召回 Quick Link)
   - 基于物品的协同过滤（ItemCF）
   - Swing模型
   - 基于用户的协同过滤（UserCF）
   - 离散特征处理
   - 矩阵补充
   - 双塔模型：模型和训练
   - 双塔模型：正负样本
   - 双塔模型：线上服务
   - 双塔模型+自监督学习
   - Deep Retrieval 召回
   - 其它召回通道
   - 曝光过滤 (Bloom Filtering)
3. **排序** (排序 Quick Link)
   - 多目标排序模型
   - Multi-gate Mixture-of-Experts (MMoE)
   - 预估分数融合
   - 播放时长建模
   - 推荐系统的特征
   - 粗排三塔模型
4. **交叉结构** (特征交叉 Quick Link)
   - Factorized Machine (FM)
   - Deep & Cross Network (深度交叉网络)
   - LHUC
   - SENet & FiBiNET
5. **用户行为序列** (用户行为序列 Quick Link)
   - 用户行为序列特征
   - DIN 模型
   - SIM 模型
6. **多样性重排** (重排 Quick Link)
   - 多样性的度量
   - MMR 算法
   - 规则约束
   - DPP：数学基础
   - DPP：多样性算法
7. **物品冷启动** (物品冷启动 Quick Link)
   - 评价指标

- 简单的召回通道
- 聚类召回
- Look-Alike人群扩散
- 流量调控
- 冷启动的AB测试

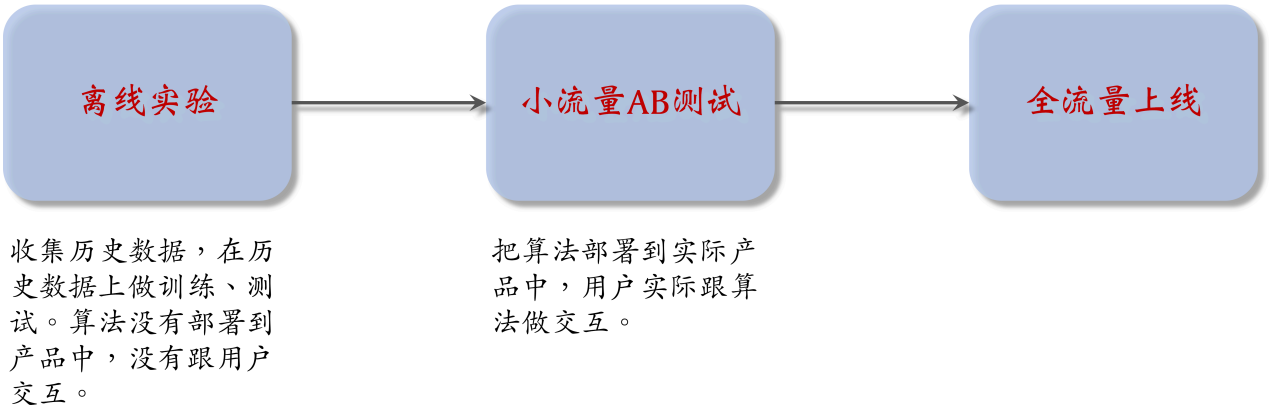
8. **涨指标的方法** (涨指标的方法 Quick Link)
   - 概述
   - 召回
   - 排序
   - 多样性
   - 特殊人群
   - 交互行为

> **额外资源**
>
> - WSS GitHub 链接
> - WSS 推荐系统公开课 (b站)
>
> 同时，个人建议结合笔记和ppt共同学习。

I. 概要

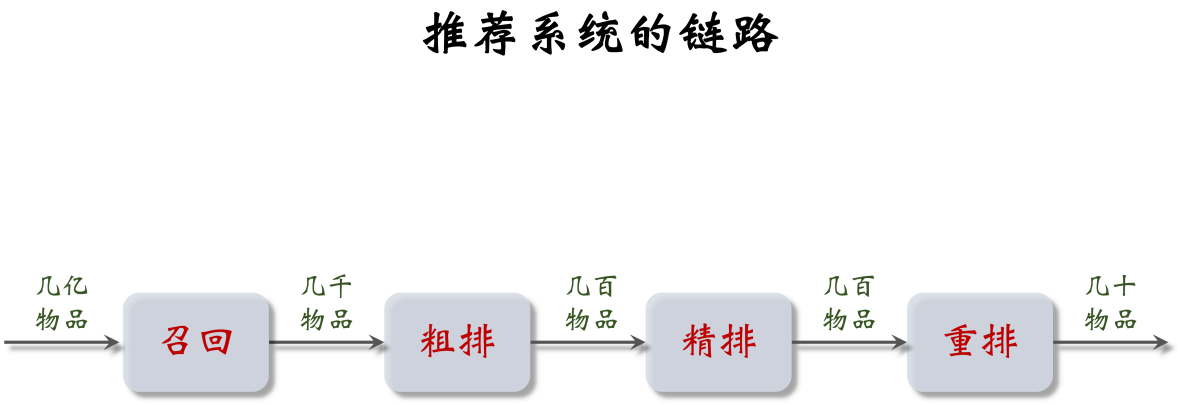
1. 小红书推荐系统概述与指标体系

- **转化流程漏斗 (Conversion Funnel):** * 用户的标准互动链路为：曝光 (Impression) → 点击 (Click) → 滑动到底 (ScrollToEnd) → 评论 (Comment)。
  - 用户点击笔记进入详情页后，也可能产生点赞 (Like)、收藏 (Collect)、转发 (Share) 等并列的强交互行为。
- **核心消费指标 (Consumption Metrics):**
  - 点击率 (CTR) = 点击次数 / 曝光次数。
  - 点赞率 / 收藏率 / 转发率 = 对应交互次数 / 点击次数。
  - 阅读完成率 = 滑动到底次数 / 点击次数 f(笔记长度)。
- **北极星指标 (North Star Metrics):** 业务增长的终极导向，主要涵盖三大维度：
  - **用户规模:** 日活跃用户数 (DAU)、月活跃用户数 (MAU)。
  - **消费属性:** 人均使用推荐的时长、人均阅读笔记的数量。
  - **发布属性:** 发布渗透率、人均发布量。
- **算法实验推进流程:** 离线实验（利用历史数据做训练和测试，无用户交互） → 小流量 A/B 测试（算法部署到实际产品中，用户真实交互） → 全流量上线。



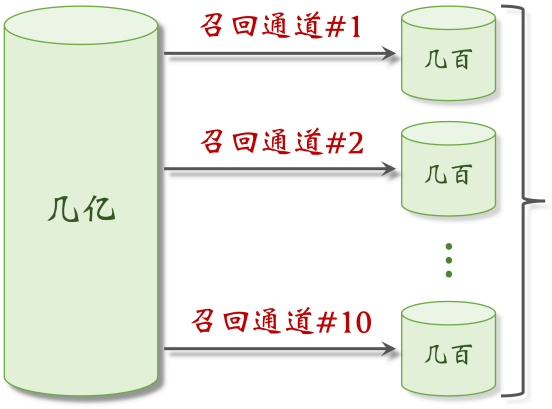
2. 推荐系统链路 (Recommendation System Pipeline)

- **多阶段漏斗架构:**



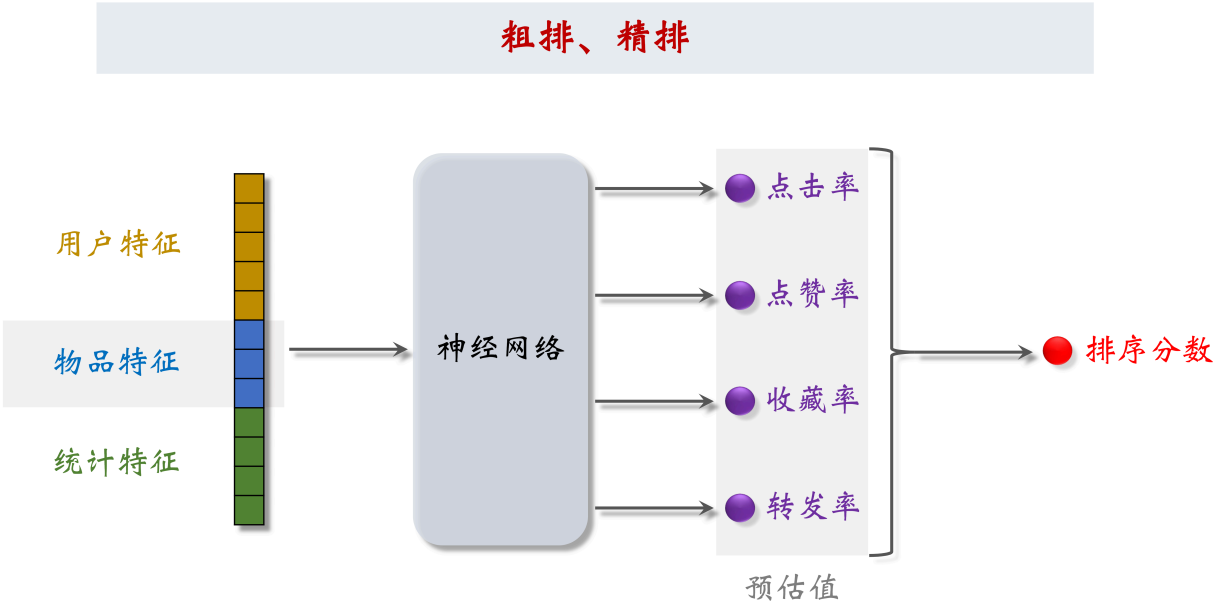
- **召回 (Retrieval):** 从几亿的海量物品池中，通过多条独立通道（如协同过滤、双塔模型、关注作者等）快速取回几千篇笔记。其核心要求是高并发、低延迟和高召回率。

> 召回通道：协同过滤、双塔模型、关注的作者、等等。



- **粗排 (Pre-ranking):** 采用小规模神经网络对召回的几千篇笔记进行快速打分，筛选出分数最高的几百篇，起到精简算力负担的作用。

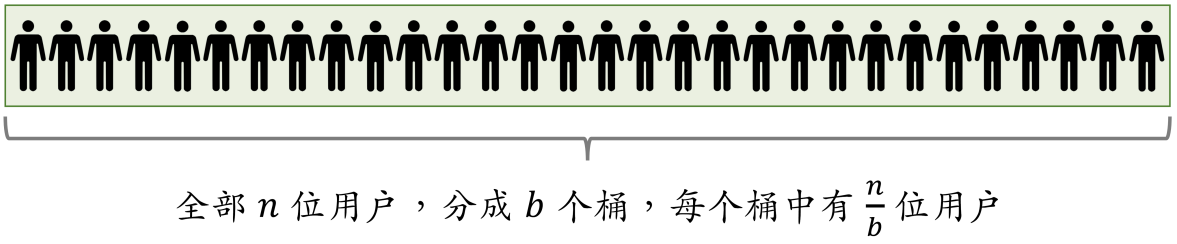
- **精排 (Ranking):** 采用大规模深度神经网络，对粗排剩留的几百篇笔记进行多指标（点击、点赞、收藏率等）的精准预估，并融合成最终的排序分数。



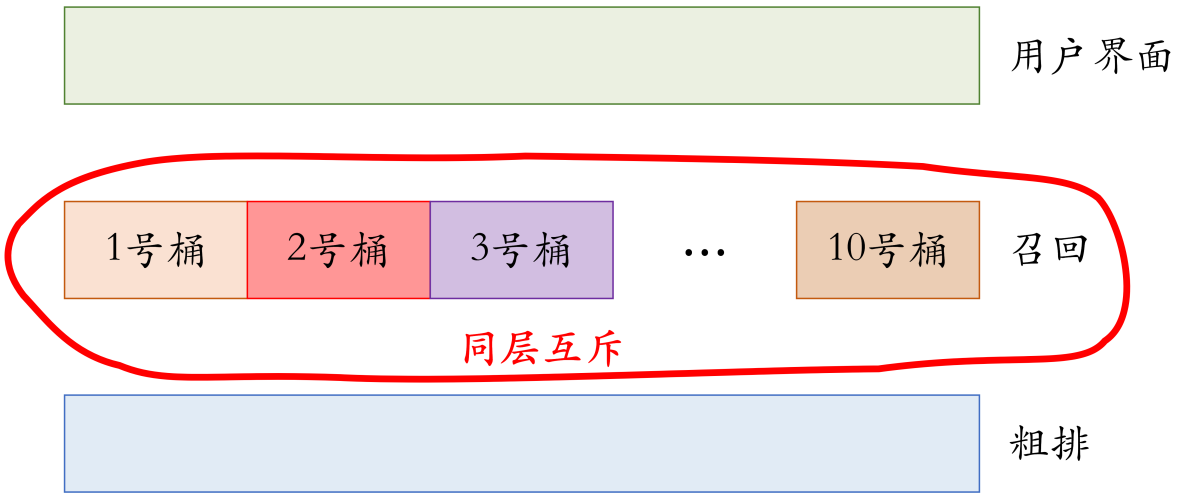
- **重排 (Re-ranking):** 负责最终的用户体验把控。主要进行多样性抽样（如 MMR 或 DPP 算法分散相似笔记）、执行规则打散，并根据生态与商业化要求插入广告和运营内容，最终将几十篇笔记展现给终端用户。

3. A/B测试机制 (A/B Testing Mechanism)

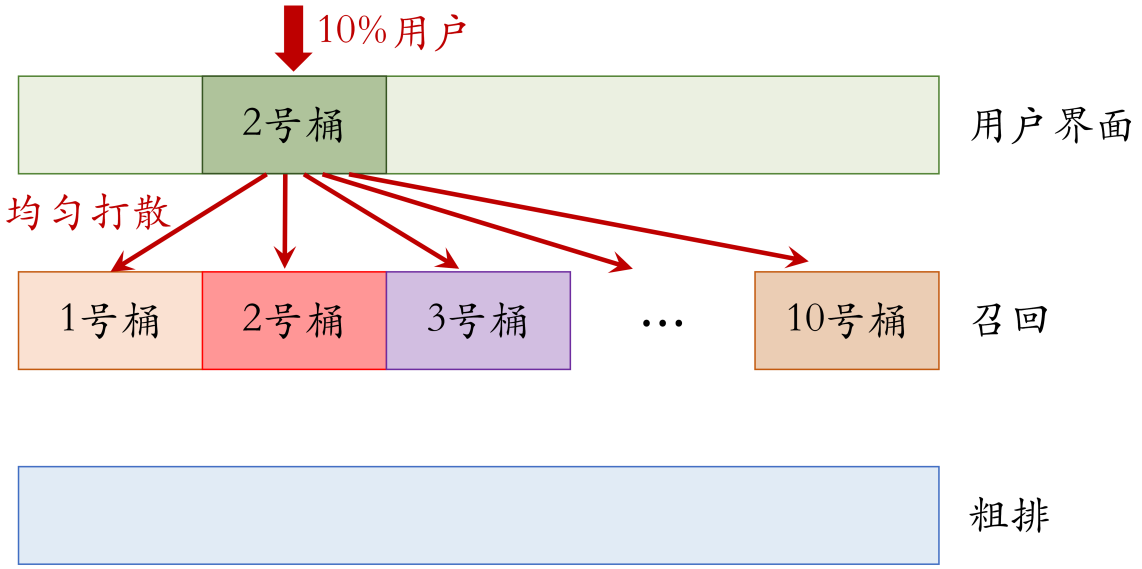
- **随机分桶 (Random Bucketing):** 利用高效哈希函数（如对用户 ID 与实验名称加盐求模）将线上用户均匀打散（例如划分为 10 个桶，每桶占 10% 流量），确保各桶用户在基线指标上完全齐质。



- **分层实验设计 (Layered Experimentation):**
  - **同层互斥:** 在同一物理层内划分相互独立、互斥的实验桶。一个用户在同一层中只能进入一个实验，防止同类策略相互干扰。



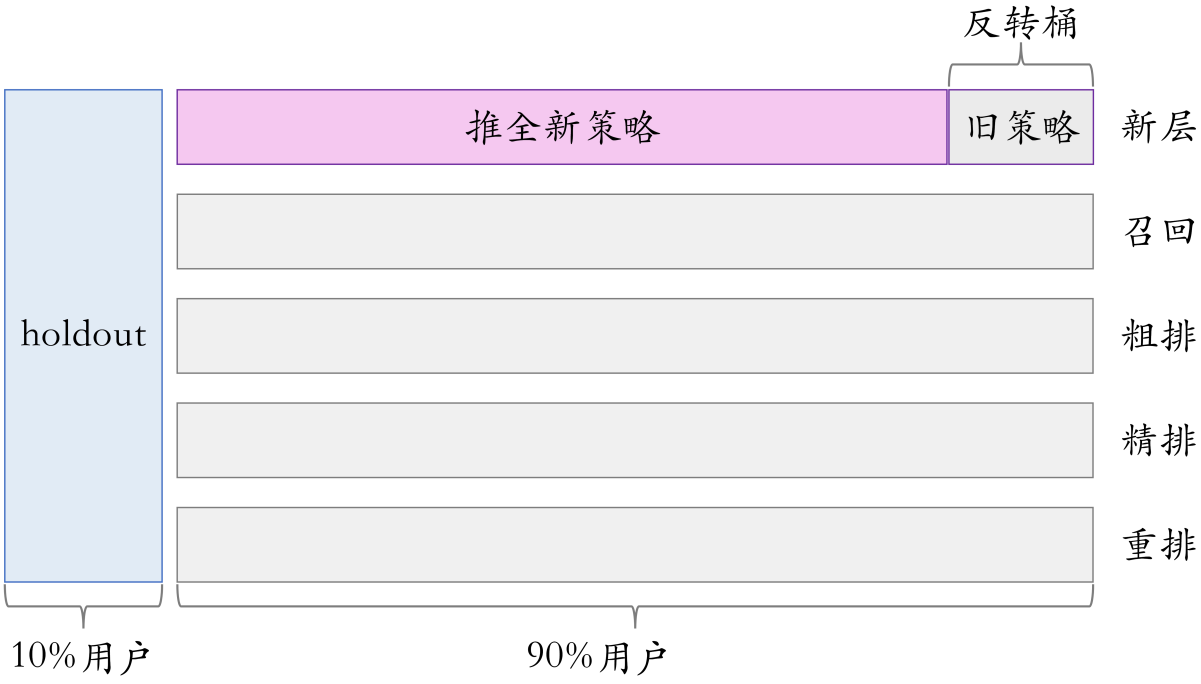
- **不同层正交:** 将召回、粗排、精排、重排等不同模块分布在不同实验层。各层之间流量相互正交（重新哈希打散），这意味着一个用户可以同时参与多个不同层的实验，实现测试流量的高效复用。



- **Holdout 机制:** 长期保留一部分绝对不受任何新实验策略影响的基准用户（如 10% 流量）。每个考核周期结束后清除旧桶并重新划分，用以客观评估整个技术团队对核心业务指标的长效总贡献。



- **实验推全与反转实验 (Reverse Experiment):** 新策略显现显著正向收益后需尽快推全以腾出实验桶。但由于部分长效指标（如留存率）具有滞后性，系统会在推全的新层中开辟一个较小的旧策略反转桶（如 10% 流量），用以长期观测和对比新旧策略的持续 diff。



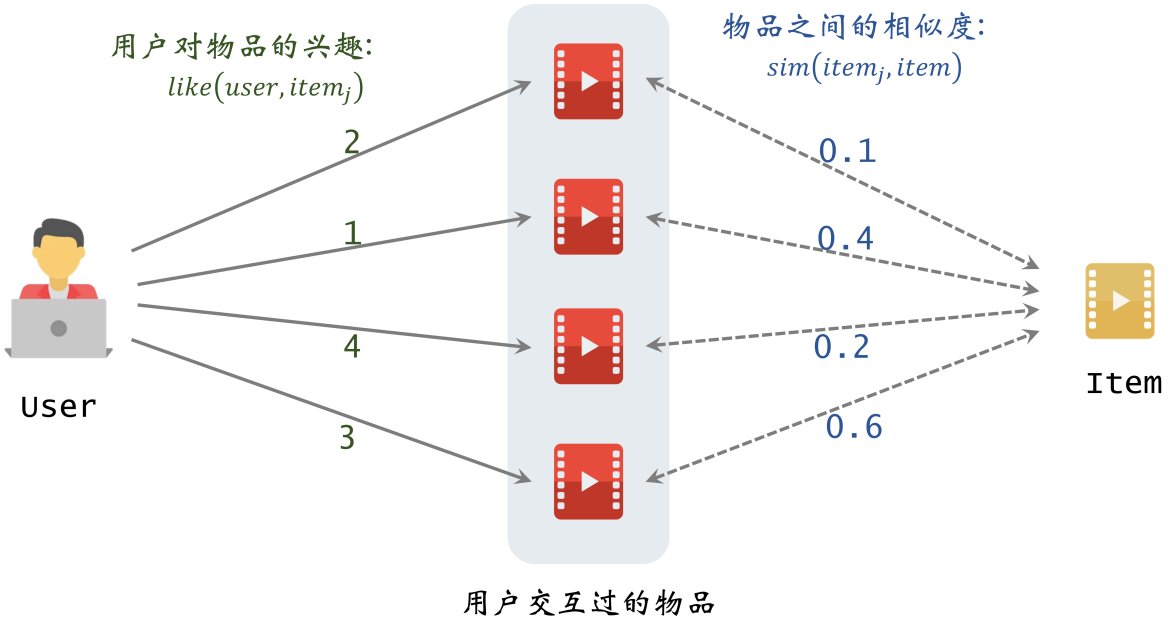
💡 Key Takeaways

- **核心指标(Key Metrics)：** 小红书推荐系统围绕“曝光 → 点击 → 互动/滑动到底”的完整转化漏斗，以用户规模（DAU/MAU）、消费时长及发布渗透率作为北极星指标，所有算法迭代均服务于这些长期核心业务价值。
- **多级漏斗架构(System Structure)：** 推荐链路采用“召回 → 粗排 → 精排 → 重排”的经典四阶段架构，将千万级物品池层层精简至几十篇笔记，在算法精度（算力消耗）与响应延迟（工程时效）之间取得了极致平衡。
- **严谨实验科学(A/B Test)：** 线上 A/B 测试通过“同层互斥、异层正交”的分层实验设计实现了流量的最大化复用，并借助长期 Holdout 机制与反转实验，科学、客观地度量策略对大盘指标的真实长效贡献。

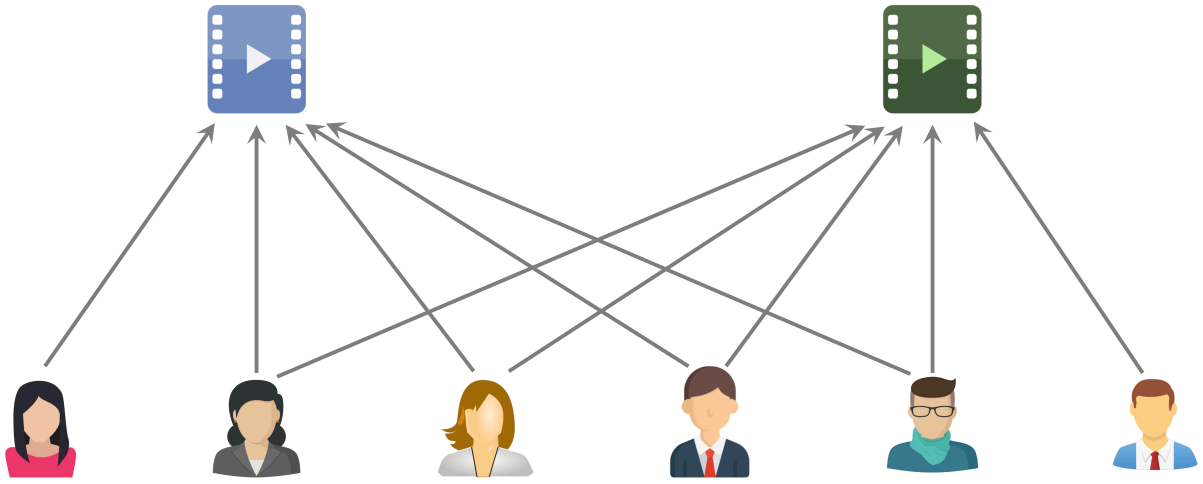
II. 召回

1. 传统召回算法：`[[协同过滤与深度召回]]` (Collaborative Filtering)

- **ItemCF (基于物品的协同过滤):**
  - **核心原理:** 如果用户喜欢物品 ie_i，且物品 ie_i 与 ie_j 在行为表现上高度相似，则该用户大概率也会喜欢 ie_j。



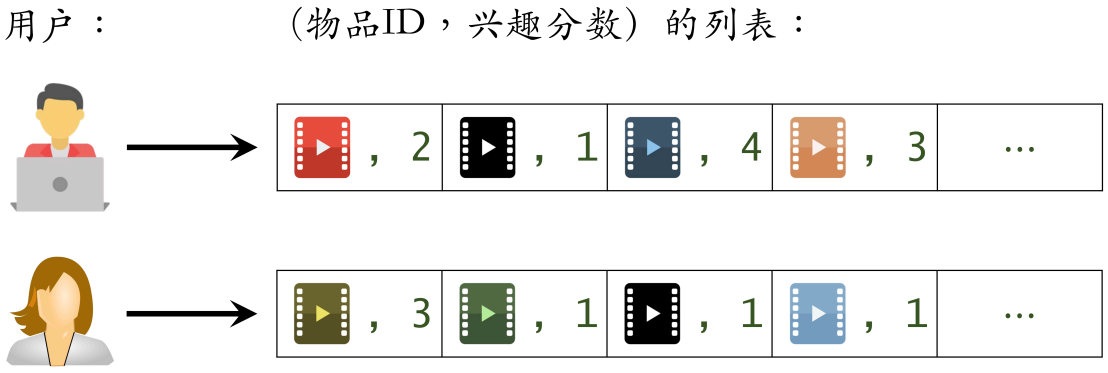
- **相似度衡量:** 将物品 (ie_i) 表示为用户交互历史的稀疏向量，相似度 $si(ie_i\, ie_j)$ 通常通过两向量夹角的余弦值（Cosine Similarity）进行离线计算。



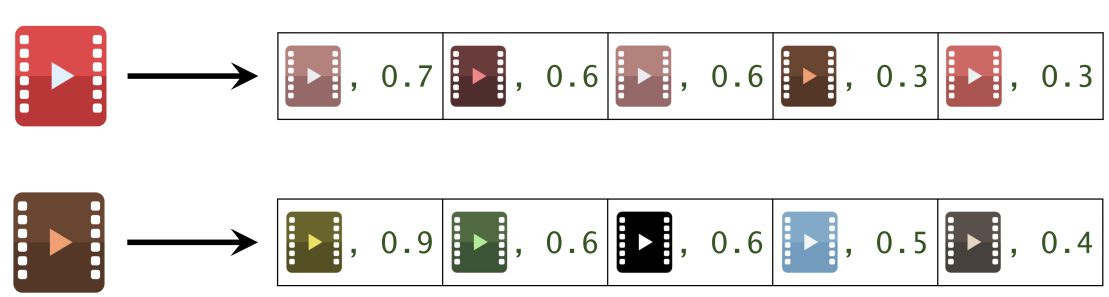
- **兴趣预估公式:**

$$\text{Score} = \underset{ior}{\text{e}}(use\, ie_i)\ \ si(ie_i\, ie_j)$$

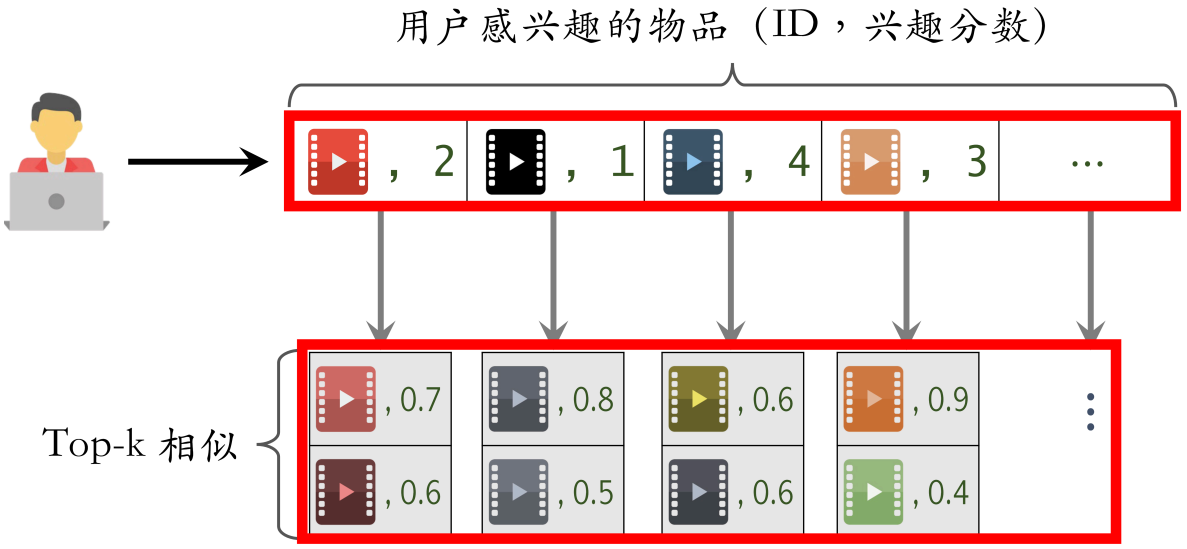
- **ItemCF 召回完整流程:**
  1. 事先做离线计算:
     - 建立“用户→物品”的索引。



     - 建立“物品→物品”的索引。



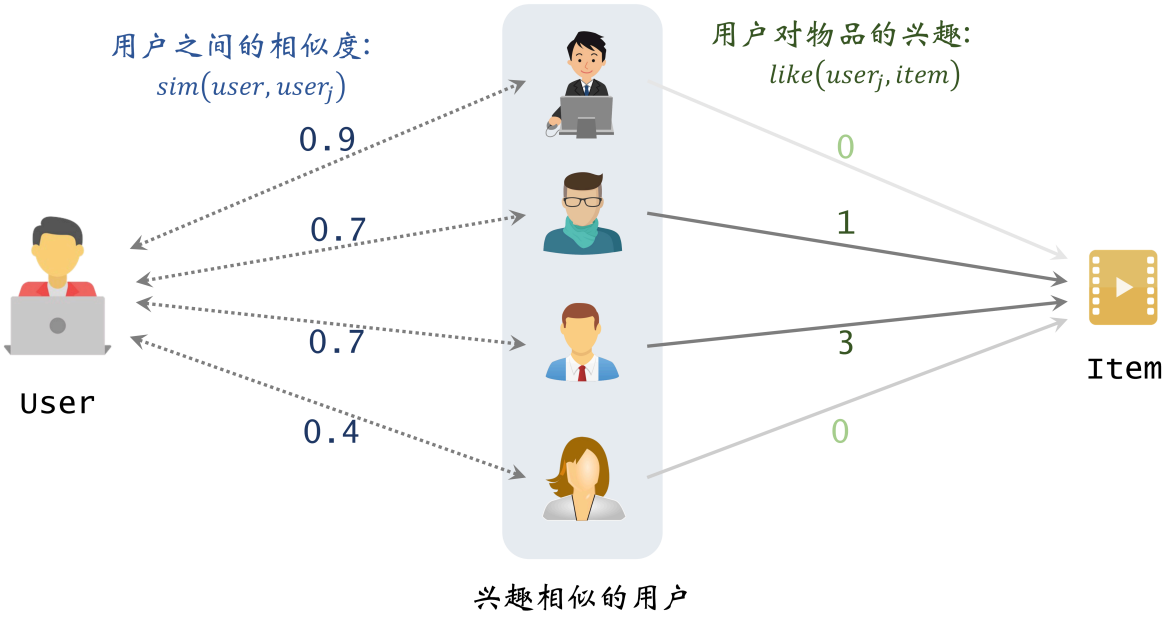
2. 线上做召回：



- **Swing 召回通道:** 作为 ItemCF 的经典升级版，在计算物品相似度时引入了“用户三元组”重叠惩罚。它通过降低共同死忠粉或热门物品的权重，有效解决了传统 CF 虚高相似度的问题。
  - **核心思路：** 额外考虑重合的用户是否来自一个小圈子。 降低共同死忠粉或热门物品的权重，有效解决了传统 CF 虚高相似度的问题。
  - **核心公式：**

    $$si(i_1\,i) = {}_{u_1u}\frac{1}{+\,eap(u_1\,u)}wee用户u_1喜欢的物品计作_1用户u喜欢的物品计作_1定义两个用户的重合度为：\ eap(u_1$$

- **UserCF (基于用户的协同过滤):**
  - **核心原理:** 如果用户 u_1 与用户 u 的兴趣高度相似，且 u 近期点击了某物品，则倾向于将该物品推荐给 u_1 。



- **核心公式：**

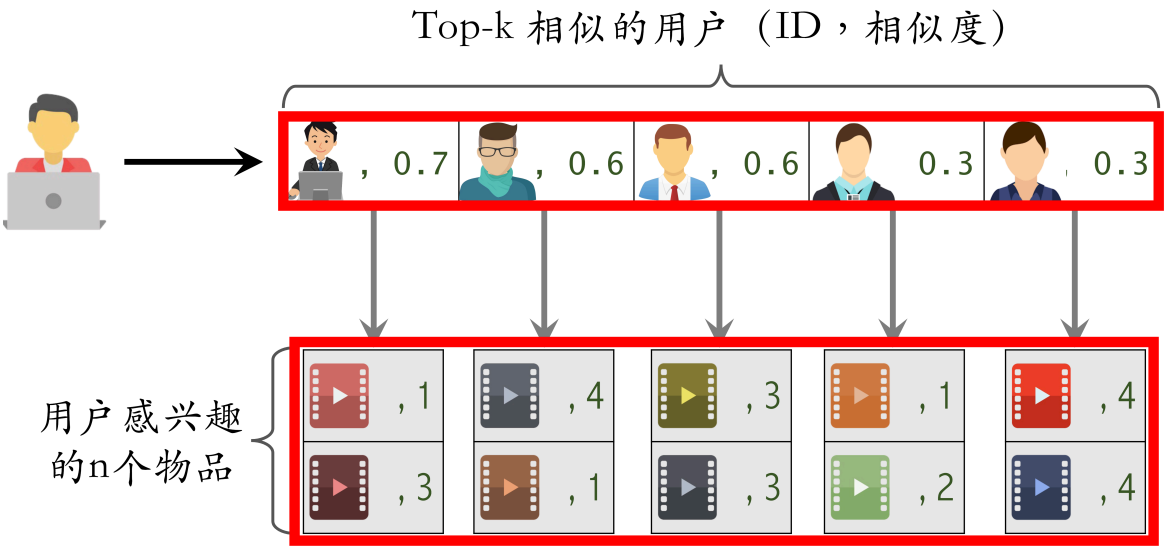
$e(use) = \quad_j si(use\,use_j)\ \ ike(use_j ie)\ wee$用户$u_1$喜欢的物品计作集合$_1$用户$u$喜欢的物品计作集合定义交集$=_1$

- **降低热门物品权重：**

用户u_1喜欢的物品计作集合$_1$用户u喜欢的物品计作集合定义交集$=_1$ 定义$n =$ 喜欢物品的用户数量，反应物品



- **线上召回流程:** 系统离线维护“用户 $\rightarrow$ 物品列表（近期交互的 n 个）”与“用户 $\rightarrow$ 相似用户列表（Top-k 个）”两个倒排索引。线上触发时，通过二级并发检索取回 $n\ k$ 个物品并计算其预估分。



2. 现代深度召回：[[双塔模型]] (Two-Tower Model)

- **离散特征处理基础:** 面对海量的高维离散特征（如用户 ID、物品 ID 等），通过 Embedding （嵌入）层将其映射为低维稠密向量。
  - **Embedding 参数量公式:** 参数量 = 向量维度 类别数量。
  - **编程实现：** TensorFlow、PyTorch提供embedding层。

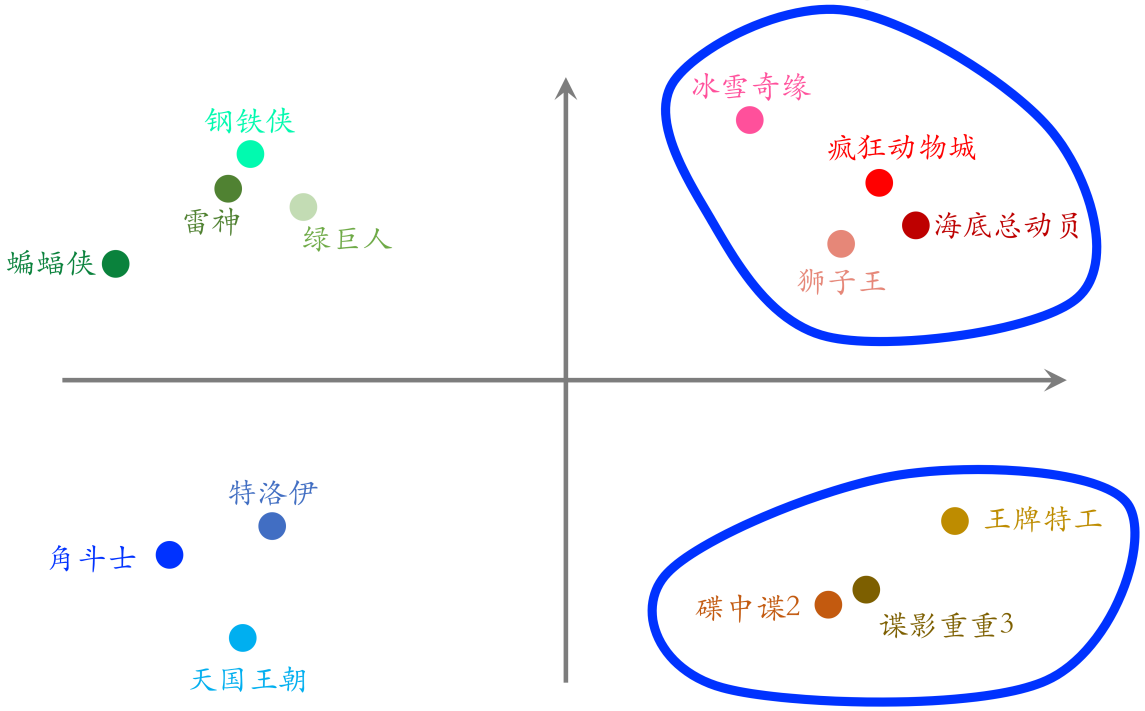
例2： 物品ID的Embedding

- 数据库里一共有 10,000 部电影。
- 任务是给用户推荐电影。
- 设 embedding 向量的维度是 16。

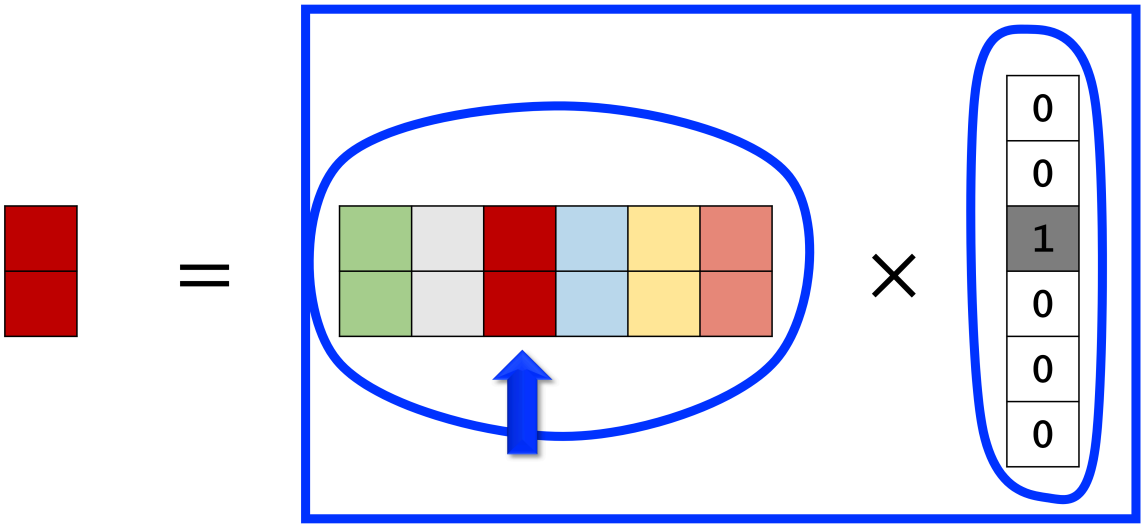
Embedding 层有多少参数？

- 参数数量 = 向量维度 × 类别数量 = 160,000

例2： 物品ID的Embedding

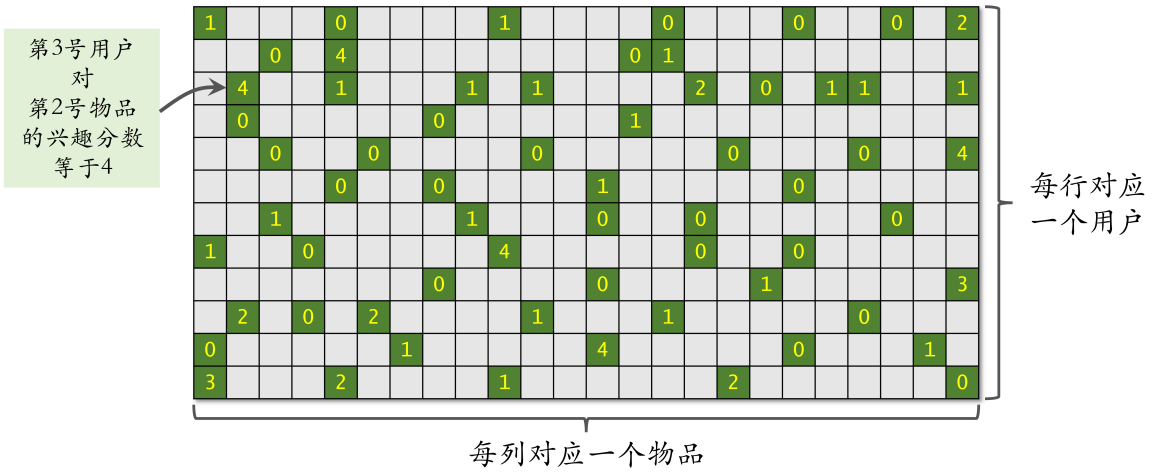


Embedding = 参数矩阵 × One-Hot向量



- **矩阵补充 (Matrix Completion):**
  - **基本想法:** 将用户embedding参数矩阵记作A。第u号用户对应矩阵第u列，计作向量a_u。将物品embedding参数矩阵记作B。第i号物品对应矩阵第i列，记作向量$_i$。我们希望训练模型学习矩阵A和B使得内积$a_{u\,i}>$是第u号用户对第i号物品兴趣的预估值。
  - **模型训练：**
    - 数据集：(用户ID，物品ID，兴趣分数)的集合，记作 $=(u\,i\,y)$。
    - 训练步骤：将用户ID、物品ID映射成向量 (i.e. 第u号用户 a_u; 第i号物品 $_i$)，并求解$in_{(uiy)}(y\ \ a_{u\,i}>)$。

绿色位置表示曝光给用户的物品；灰色位置表示没有曝光。



- 缺点：实践表现不好，需要改进。

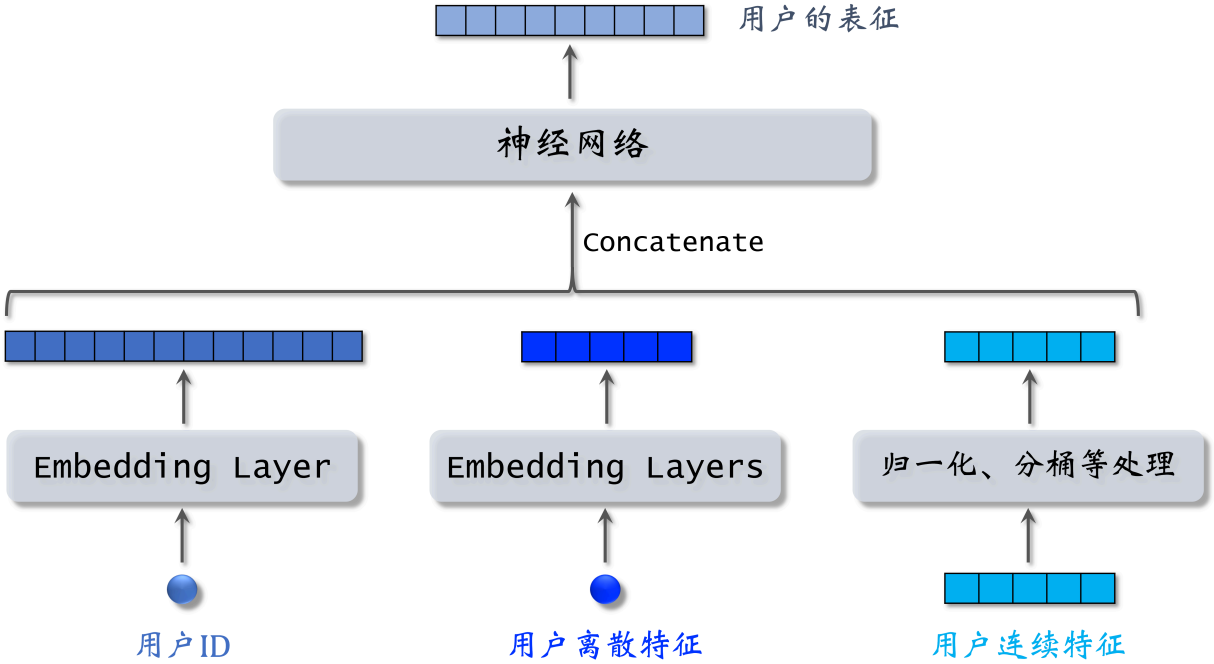
缺点1：仅用ID embedding，没利用物品、用户属性。

缺点2：负样本的选取方式不对。

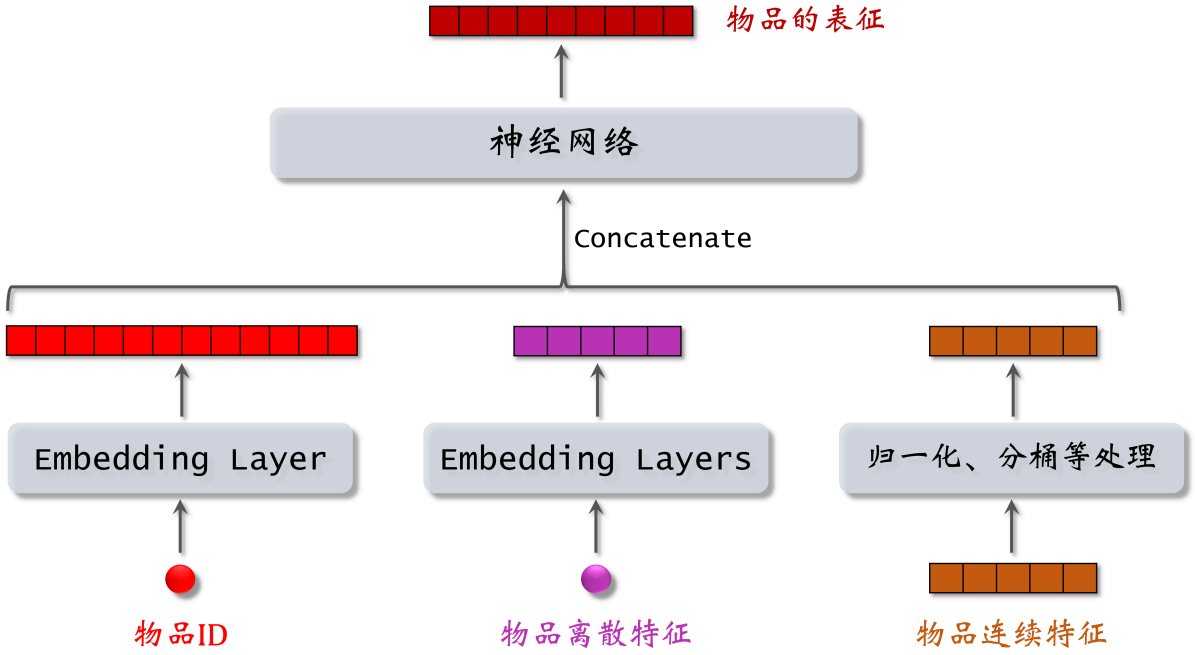
缺点3：做训练的方法不好。

- 内积 $\langle \mathbf{a}_u, \mathbf{b}_i \rangle$ 不如余弦相似度。
- 用平方损失（回归），不如用交叉熵损失（分类）。

- **双塔架构设计:** 模型解耦为相互独立的**用户塔 (User Tower)** 和 **物品塔 (Item Tower)**。
  - 用户塔聚合用户 ID、离散及连续特征，通过多层全连接网络输出表征向量 。

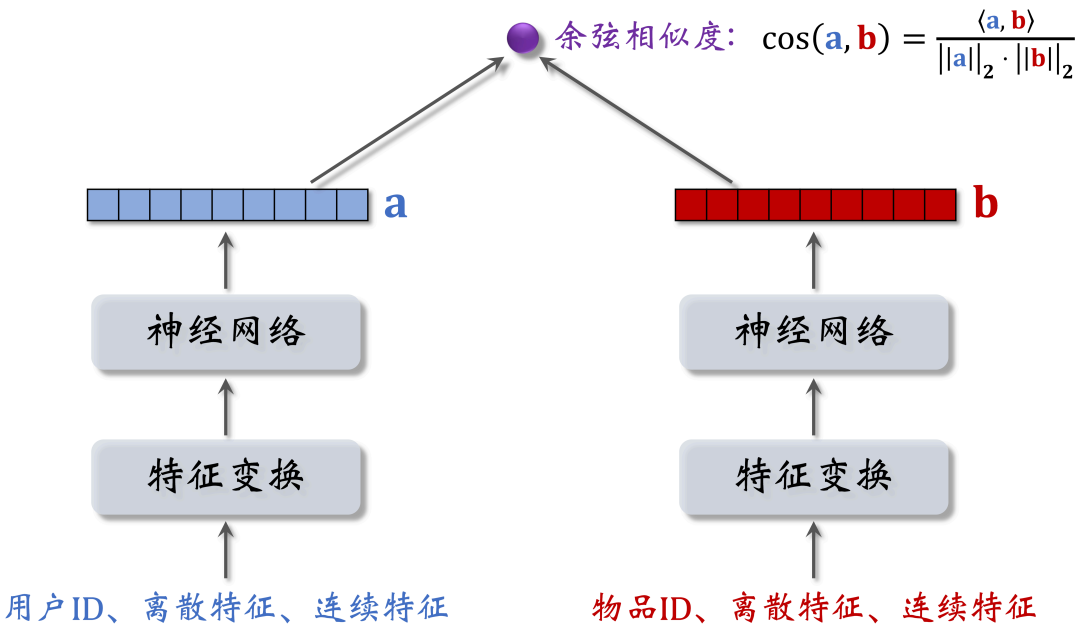


  - 物品塔聚合物品 ID、属性和连续特征，输出表征向量 。

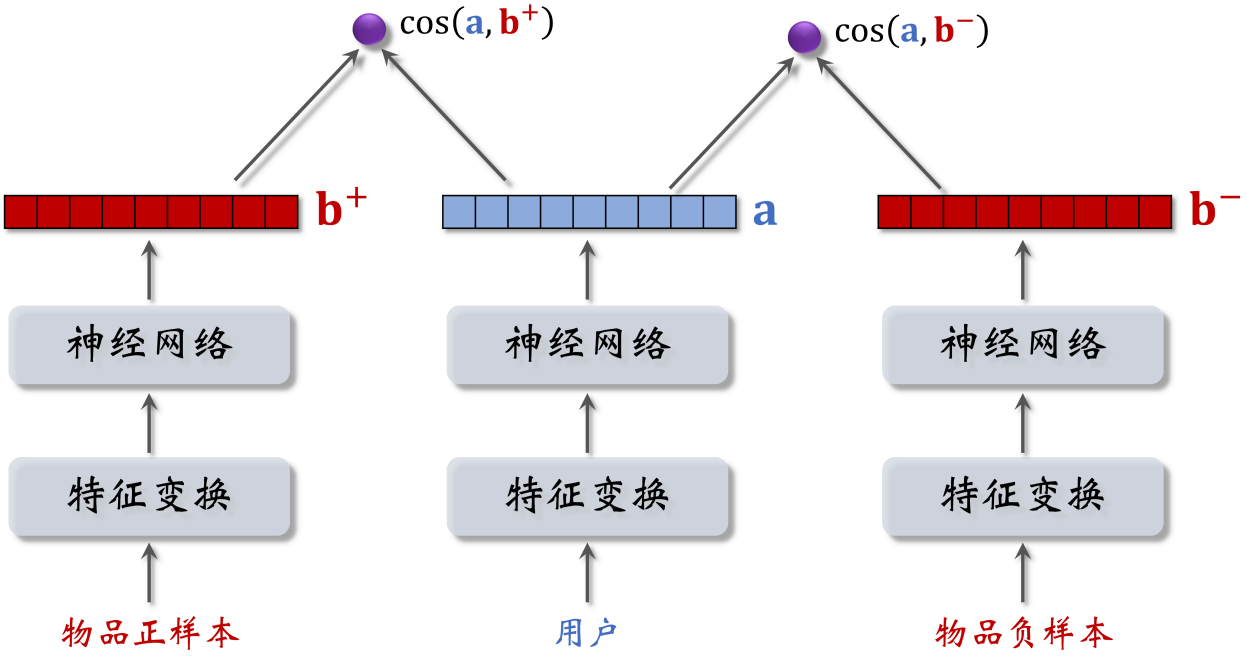


- 最终的兴趣预估分数通过两者的余弦相似度 co( ) 或内积进行表达。

双塔模型



- **双塔模型的训练：**
  1. **Pointwise:** 独立看待每个正样本、负样本，做简单的二元分类。
     - **基本想法：** 对于正样本，鼓励co( )接近1；对于负样本，鼓励co( )接近co( )接近-1。
     - 正负样本数量比例为1：2或者1：3
  2. **Pairwise:** 每次取一个正样本、一个负样本。



- **基本想法：** 鼓励co($^{+}$) 大于co( )。
- **损失函数：** 如果co($^{+}$) 大于co( ) + ，则没有损失；否则，损失等于co( ) co($^{+}$) + 。

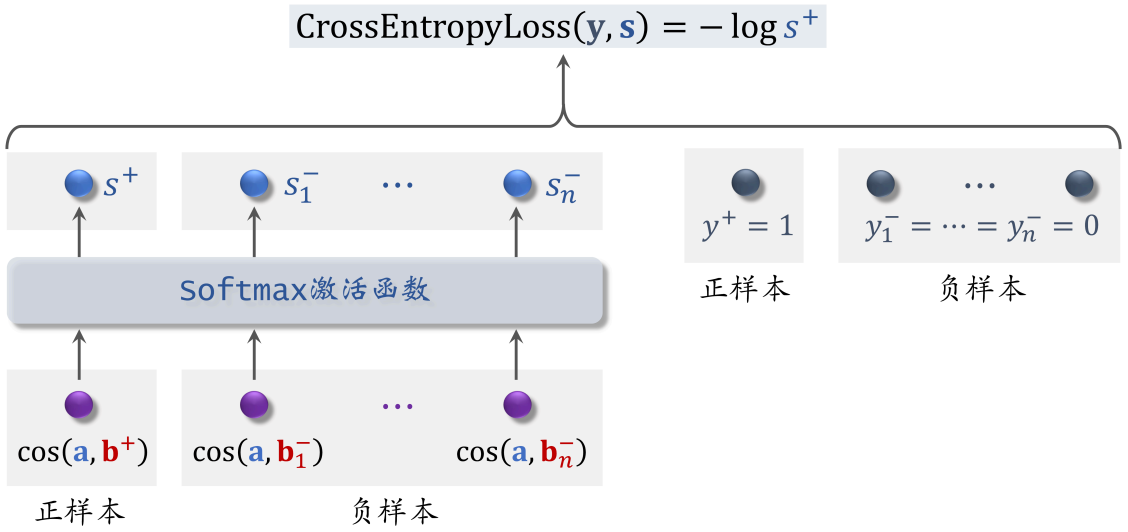
Triplet hinge loss:

$$L(\mathbf{a}, \mathbf{b}^+, \mathbf{b}^-) = \max\{0, \cos(\mathbf{a}, \mathbf{b}^-) + m - \cos(\mathbf{a}, \mathbf{b}^+)\}.$$

Triplet logistic loss:

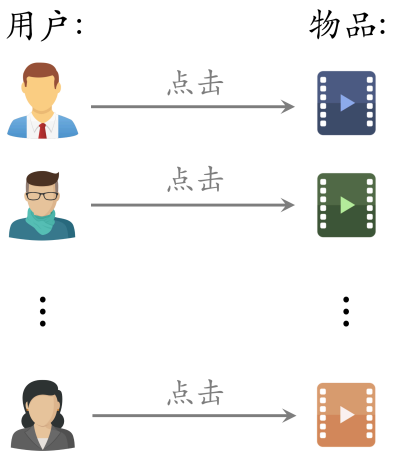
$$L(\mathbf{a}, \mathbf{b}^+, \mathbf{b}^-) = \log(1 + \exp[\sigma \cdot (\cos(\mathbf{a}, \mathbf{b}^-) - \cos(\mathbf{a}, \mathbf{b}^+))]).$$

3. **Listwise:** 每次取一个正样本、多个负样本。
   - **基本想法：** 鼓励co($^{+}$)尽量大；鼓励co($_{1}$) ... co($_{n}$)尽量小。
   - **损失函数：**



- **正负样本选择与抽样纠偏:**
  - **正样本:** 真实发生点击交互的“用户—物品”对。
  - **简单负样本：**

- **全体物品：** 未被召回的物品，大概率是用户不感兴趣的。所以，违背找会的物品全体物品。所以，我们可以从全体物品中抽样作为负样本。
- **Batch内负样本 (In-batch Negatives):** 选取同一个 batch 内其他用户的正样本物品充当当前用户的负样本。为消除由于热门物品频繁入包导致的**抽样偏差 (Sampling Bias)**，训练时需结合物品被抽样到的概率 p_j（正比于点击热度）进行减数纠偏，将预测目标调整为：$\mathrm{co}(_{i\,j})\ \mathrm{o}\,p_{j}$。
  ![[02_Retrieval_07.pdf#page=10&rect=12,44,952,452|02_Retrieval_07, p.10]]



- 一个物品出现在 batch 内的概率 $\propto$ 点击次数。
- 物品成为负样本的概率本该是 $\propto (\text{点击次数})^{0.75}$，但在这里是 $\propto$ 点击次数。
- 热门物品成为负样本的概率过大。

参考文献：

- Xinyang Yi et al. Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations. In *RecSys*, 2019.

  - **困难负样本:** 混合一部分被前级粗排或精排淘汰掉的物品（典型配比：50% 简单负样本 + 50% 困难负样本），逼迫模型学习更细粒度的决策边界。
  - ⚠️ **负样本禁区:** 严禁将线上“曝光但未点击”的物品用作召回模型的负样本（这类样本属于精排模型的负样本），否则会导致召回网络产生严重的样本选择偏差。
- **工程部署与在线查找:** 物品向量在离线计算完成后一次性推入 Milvus 或 Faiss 等向量数据库；线上召回时，仅需实时调用用户塔神经网络现算用户当前向量，并将其作为 Query 去检索向量数据库中余弦相似度最高的前 k 个物品向量。
  Notice: To see more, check 最近邻查找PPT(Retrieval05) & 线上召回和更新 (Retrieval08).

3. 路径表征召回：Deep Retrieval 架构

- **核心理念:** 传统的双塔模型依赖线上最近邻向量查找，而 **Deep Retrieval** 突破性地将物品表征为多条由根到叶的**路径 (Path)**，将大规模物品检索问题巧妙转化为神经网络对路径的概率预估问题（类似于阿里的 TDM 架构）。
- **在线召回三步法:**
  1. **构建双向索引:** 离线维护 `Path → List<Item>` 和 `Item → List<Path>` 的映射索引。
  2. **Beam Search 路径检索:** 给定用户特征向量，神经网络通过 **Beam Search**（束宽如 size=4）逐层贪心筛选，检索出用户最感兴趣的几条最优路径，极大程度节省了全量路径的计算开销。
  3. **物品映射:** 根据倒排索引 `Path → List<Item>` 召回对应的候选物品，最后做简单过滤打分。
- **联合训练机制:** 在模型训练过程中，同时学习神经网络参数（增大用户点击物品对应的路径概率得分）与物品的路径表征。在更新物品路径映射时，为了防止过多物品扎堆在极少数热门路径上导致严重的线上计算负载倾斜，目标函数中加入了惩罚正则项 $\mathrm{re}(pa)$ = 该路径上物品数量。
- Notice: 此方法了解即可。

4. 辅助召回通道与 `[[特征处理与曝光过滤]]` (Bloom Filter)

- **其他特色召回通道:**
  - **关注/有交互/相似作者召回:** 围绕创作者生态，拉取用户已关注作者、曾发生强交互作者或其相似作者发表的最新笔记。
  - **缓存召回 (Cache Retrieval):** 复用前几次推荐中进入精排前 50 名但因重排打散未获得实际曝光的笔记。设置固定容量的缓存和退场机制（如曝光退场、最多召回 10 次、最长保存 3 天），大幅挽回精排的算力

浪费。
    - **地理位置召回:** 基于 GeoHash 或同城经纬度定位，召回距离当前用户物理距离最近的社区笔记。
- **实时曝光去重必备 —— Bloom Filter (布隆过滤器):**
    - **应用背景:** 必须在线上实时过滤掉已经向该用户曝光过的物品，以防止出现内容重复推荐，劣化用户观感。
    - **运行原理:** 维护一个超长二进制向量（维度为 ）和 k 个独立的随机哈希函数。每当某物品成功曝光给用户，便通过 k 个哈希函数计算其索引位置，并将二进制向量中对应的 k 个位强行置为 1。
    - **误伤概率 (False Positive Rate):** 随着已曝光物品集大小 n 不断增大，向量中 1 的占比越来越高，未曝光物品被误判为已曝光的概率也会单调上升，公式近似为：

$$1 \ \mathrm{e} \left(\frac{kn}{}\right)$$

    - **工程权衡:** 越大向量越长，越不易发生哈希碰撞；k 的取值需要根据可容忍的误伤概率门槛计算出理论的最优解。

💡 Key Takeaways

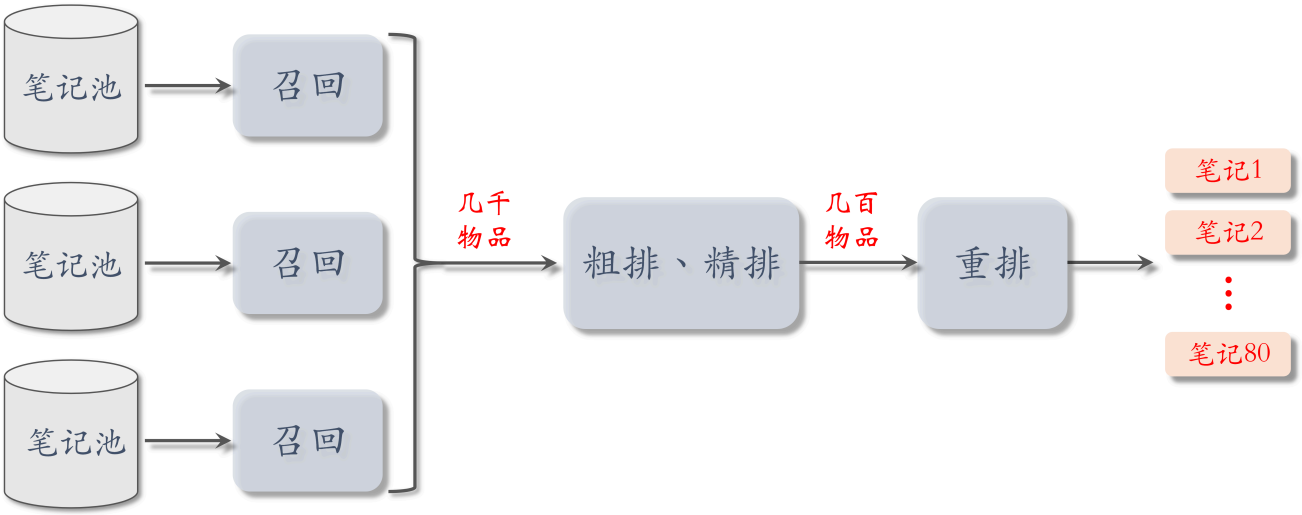
- **算法范式演进：** 召回技术经历了从传统协同过滤（基于行为重合度的 ItemCF、UserCF 及引入三元组惩罚的 Swing）向深度学习（双塔模型、基于路径表征的 Deep Retrieval）的跃升，实现了对海量高维离散特征的强语义表征。
- **工程与样本解耦：** 现代深度召回（如双塔模型）通过将用户和物品解耦、离线构建 Faiss/Milvus 向量库，实现了线上高并发的最近邻检索；而在训练中，采用 In-batch 负样本配合热度纠偏（$\mathrm{o}\, p_j$），且**严禁使用曝光未点击样本作为召回负样本**，是保障召回效果的关键。
- **生态扩展与去重保障：** 系统通过引入关注、同城和精排缓存等辅助通道丰富了推荐的多元生态，同时在线上实时利用布隆过滤器（Bloom Filter）在有限的空间复杂度下完成了超大规模的曝光去重，有效避免了内容的重复推荐。

III. 排序

1. 推荐系统链路与交互指标

推荐链路概览

- **工业级推荐四阶段：** 推荐链路包含“笔记池 $\rightarrow$ 召回 $\rightarrow$ 粗排/精排 $\rightarrow$ 重排”的完整多级过滤漏斗 。
- **物品池规模逐层递减：** 召回阶段将千万级物品池初筛至几千件物品；粗排与精排阶段将其进一步精简至几百件物品；重排处理后最终向用户展示约80篇笔记 。



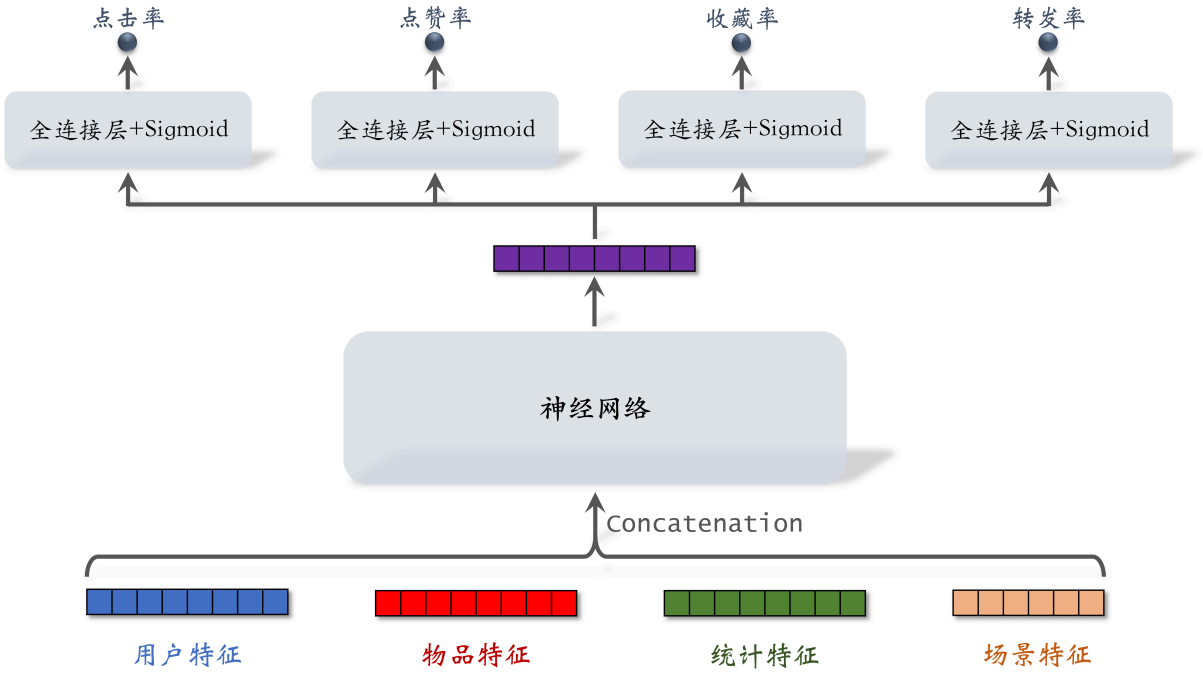
用户与笔记的交互指标

- **核心交互记录：** 系统实时记录每篇笔记的五大核心表现指标：曝光次数（Impressions）、点击次数（Clicks）、点赞次数（Likes）、收藏次数（Collects）和转发次数（Shares）。
- **核心转化率公式：** 各项互动率均以点击次数（而非曝光）作为分母进行条件概率建模：
  - 点击率() $= \frac{\text{点击次数}}{\text{曝光次数}}$
  - 点赞率 $= \frac{\text{点赞次数}}{\text{点击次数}}$
  - 收藏率 $= \frac{\text{收藏次数}}{\text{点击次数}}$
  - 转发率 $= \frac{\text{转发次数}}{\text{点击次数}}$
- **排序依据：** 多目标模型同时预估上述多种指标分数，并通过融分公式（如加权和、连乘等）合成最终价值分数，以此进行统一排序与截断。

2. 多目标排序模型架构与训练

基础多任务网络架构（Shared-Bottom）

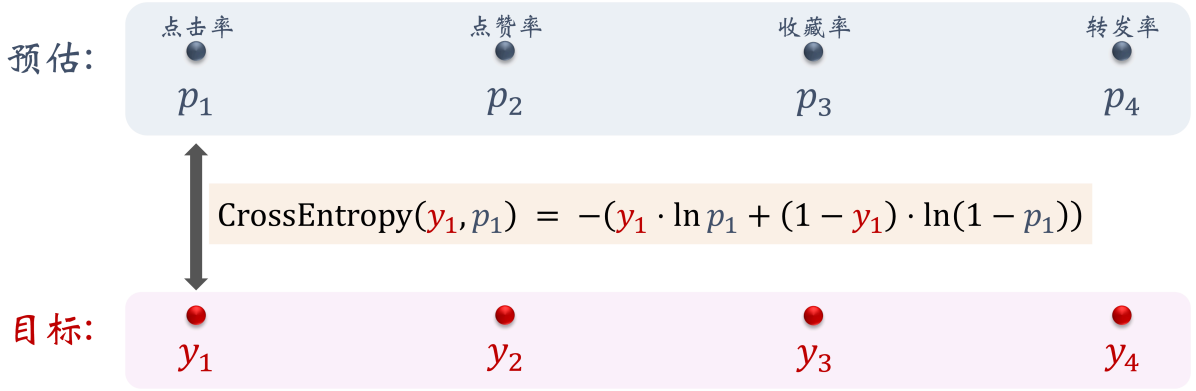
- **前期融合（Early Fusion）：** 底层将用户特征、物品特征、统计特征和场景特征直接进行拼接（Concatenation），统一输入给共享的底层神经网络。



- **多任务输出塔：** 共享网络的顶层表征向量分别进入相互独立的“全连接层 + Sigmoid”任务塔分支，并行预估点击率（p_1）、点赞率（p）、收藏率（p_3）及转发率（p） 。

损失函数与预估值校准

- **交叉熵损失（Cross Entropy）：** 每个预估任务塔都采用二元交叉熵作为单任务损失函数：



$$\text{roro}(y_i\, p_i) = y_i \cdot\ p_i\ \ (1\ \ y_i) \cdot (1\ \ p_i)$$

- **联合训练目标：** 整个多目标模型的总损失函数为各个任务损失的加权和：$_{i\,i} \cdot \text{roro}(y_i\, p_i)$ 。
- **负样本降采样（Down-sampling）：** 推荐系统中未点击、未互动样本占比极高（类别不平衡），训练时会对负样本进行降采样以平衡正负样本比例，大幅节约计算算力 。
- **线上预估值校准：** 由于训练时负样本被人为抛弃，会导致模型产出的预估值显著高于真实转化率。线上必须执行数理校准（其中 $(0\,1)$ 为负样本采样率） ：

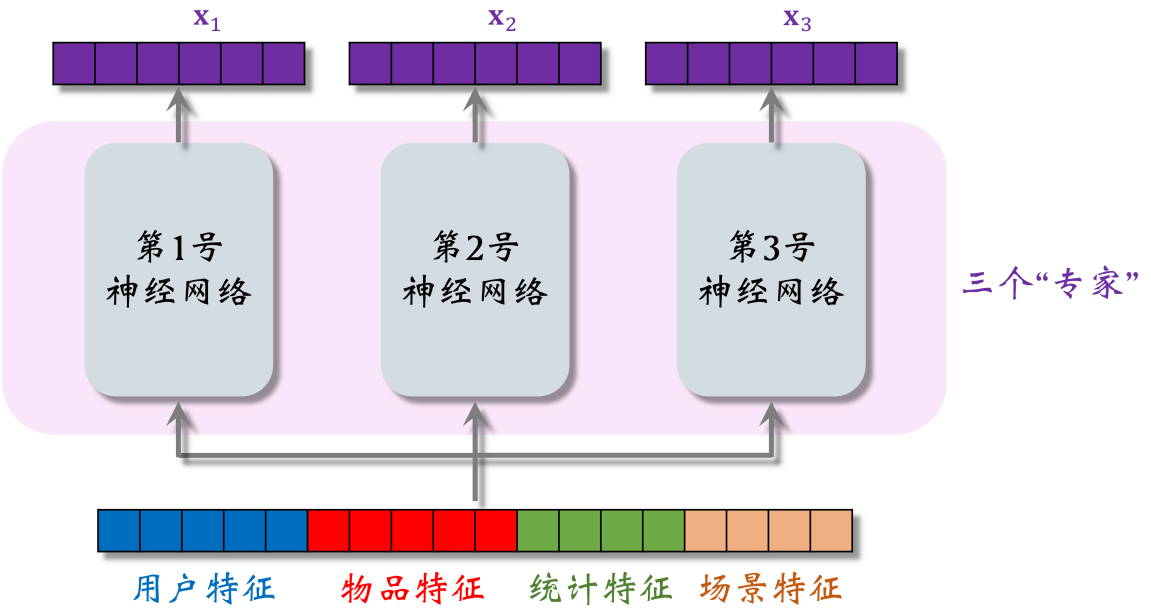
$$p_{\text{re}} = \frac{\cdot\, p_{\text{re}}}{(1\ ) \cdot p_{\text{re}} +}$$

> **Notice:**
>
> 有的同学也许会注意到，我们先负样本降采样，再校准预估值，不是南辕北辙了吗？实则，不然。我们不难注意，负样本降采样是为了使我们的模型更好学习正样本特征，否则过度估计负样本同样可以降低损失函数，倒是拟合误差较大。

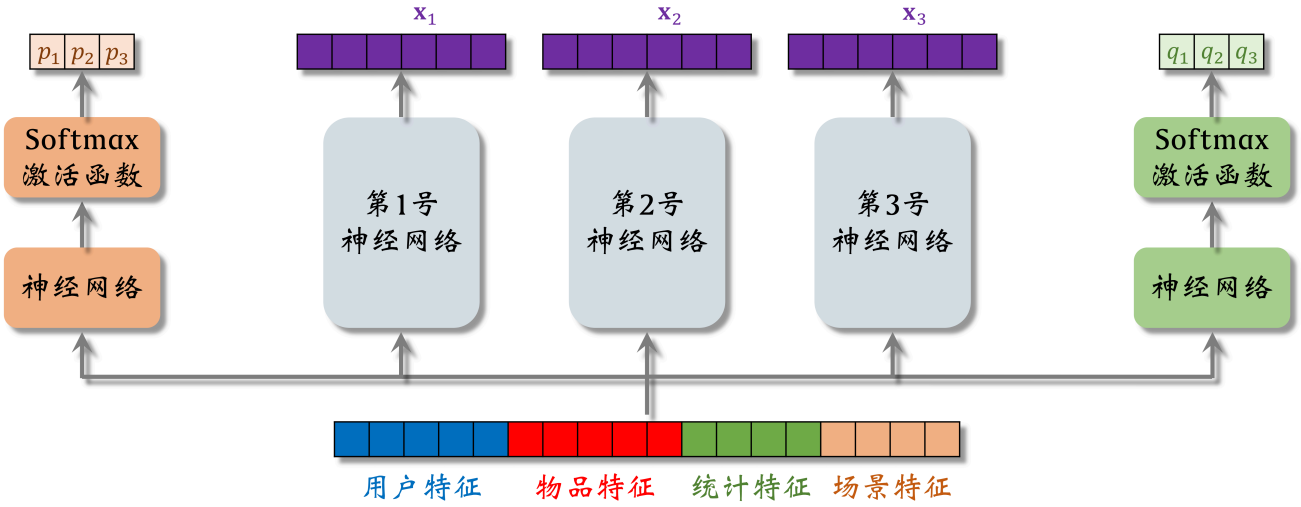
3. MMoE模型与极化问题

MMoE (Multi-gate Mixture-of-Experts) 架构

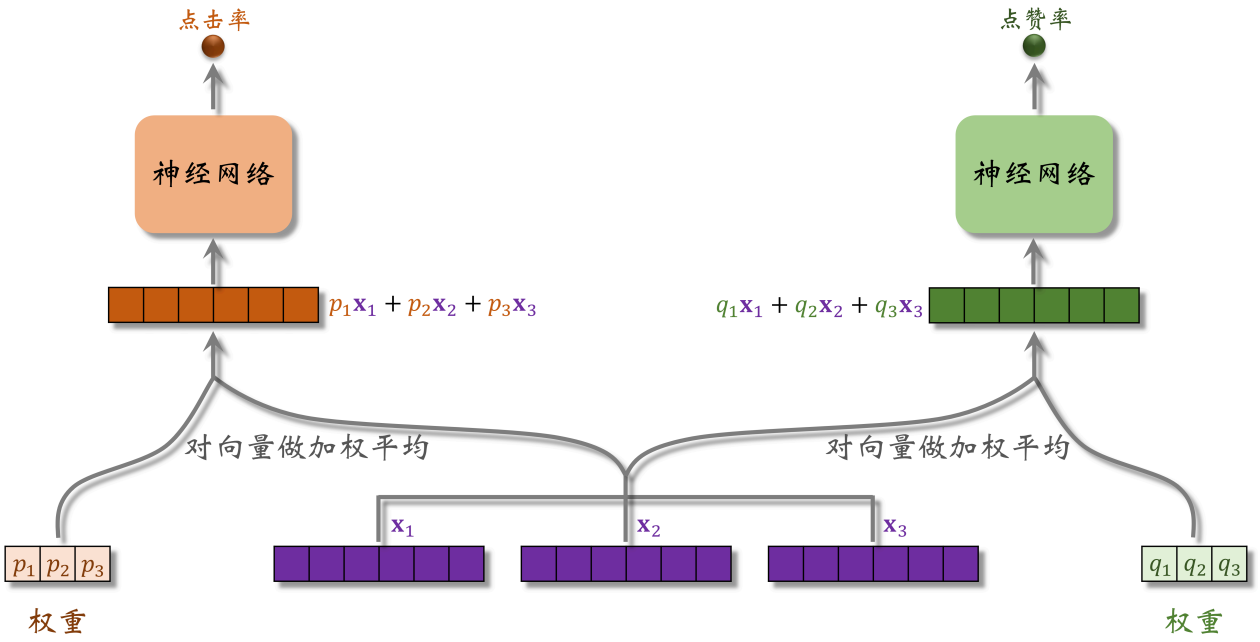
- **打破底层共享瓶颈：** MMoE 将底层的单一共享网络拆解为多个并行的“专家神经网络”(Experts)，允许不同专家学习不同侧面的特征表征 。



- **多门控网络机制（Multi-gate）：** 每个预测任务单独配置一个专属的门控网络（Gating Network），通过输入特征经过神经网络并由 Softmax 激活，动态计算当前任务对各个专家的选择权重 。

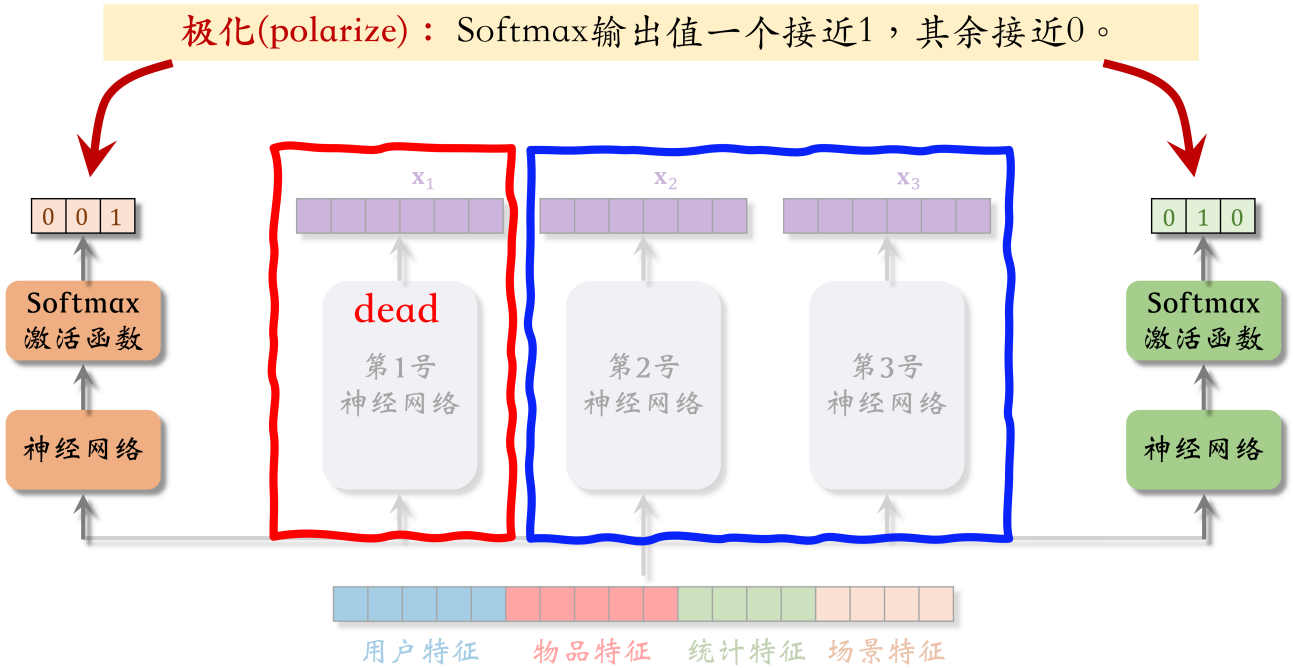


- **加权特征合成：** 任务塔的输入不再是生硬的底层拼接，而是所有专家输出向量的加权平均值（p_{ii}），实现了多任务之间参数共享与定制表征的极致平衡 。



门控网络极化（Polarization）现象与解决

- **极化定义：** 在训练中，Softmax 往往趋向于产生“赢者通吃”的现象（某一专家权重接近 1，其余接近 0），导致其他专家由于无法更新参数而退化或“死掉”（dead expert） 。



- **Dropout 解决方案：** 在训练期间对 Softmax 的输出应用 Dropout（通常有 10% 的概率随机掩码掉某些专家权重），强迫其他专家承接梯度，从而有效抑制极化、激活长尾专家网络 。

4. 预估分数融合与播放时长建模

预估分数融合策略

- **简单加权和：** $= p_{\text{cc}} + w_1 \cdot p_{\text{e}} + w \cdot p_{\text{coec}} +$
- **CTR 乘积加权（点击率条件化）：** $= p_{\text{cc}} \cdot (1 + w_1 \cdot p_{\text{e}} + w \cdot p_{\text{coec}} +)$，将点击率作为大盘透传的乘数基底 。
- **融合预估分数：** $= (1 + w_1 \cdot p_{ie})^1 \cdot (1 + w \cdot p_{ike}) \cdot$
- **倒数排名融合（Rank-based Reciprocal）：** 部分短视频应用不直接使用预估分，而是将各目标的预估分数转化为序数排名 ，再通过倒数加权复合： $= \frac{w_1}{1^{\text{cc}}+1} + \frac{w}{\text{e}^+} + +$ 。

- **电商漏斗链路连乘公式：** 依据“曝光 → 点击 → 加购 → 付款”的串联转化链条进行概率连乘融合：
  $= p_{\text{1cc}}\ p_{\text{cr}}\ p_{3}\ \text{rce}$。
- 注意到，在特殊超参数下，很多融分公式有现实意义。同时，很多融分公式含有大量超参数。我们需要利用A/B测试来获取最佳超参数。

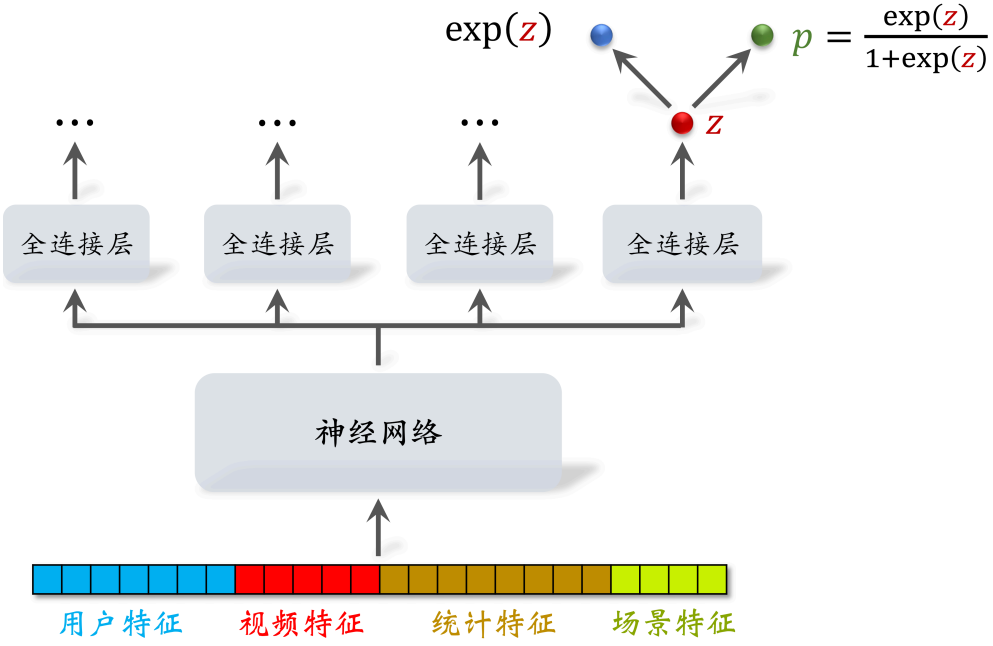
视频播放指标建模

- **YouTube 时长加权方案：** 直接采用回归拟合绝对播放时长效果较差。推荐方案是在训练时将实际观测播放时长 （未点击则 $= 0$）作为样本权重，采用加权交叉熵进行二分类训练：

$$\text{o} = \frac{}{1+} \cdot p\ \frac{1}{1+} \cdot (1\ p)$$

$$y = \frac{t}{1+t}$$

如果 $p = y$，那么 $\exp(z) = t$。



线上推理阶段，直接将倒数第二层全连接层的原始输出 执行指数变换 e()，作为绝对播放时长的期望预估值。

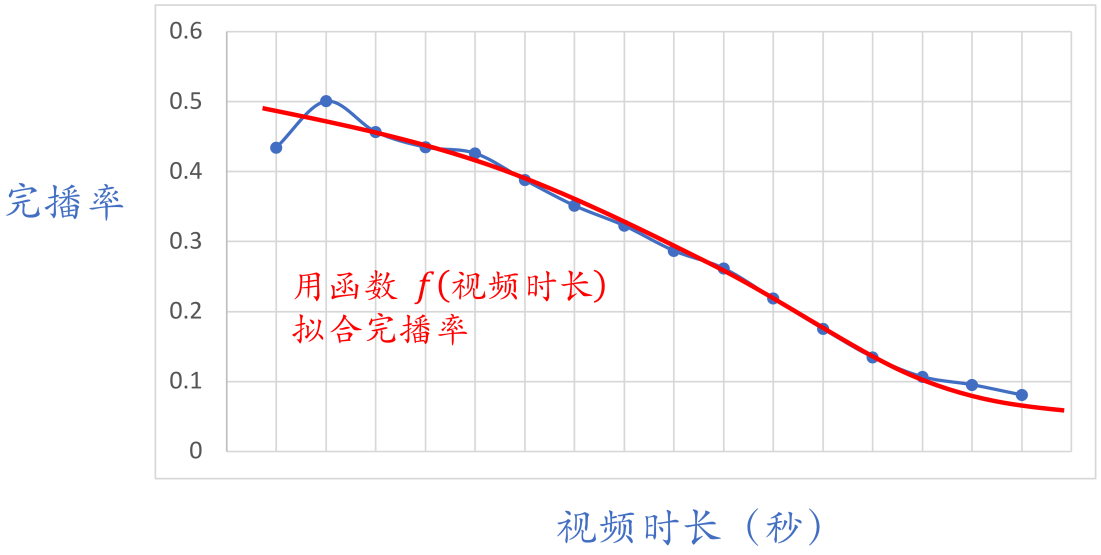
Remark:

1. 由于 并没有非负性，我们通过$ep()$变换确保非负性。
2. 我们利用sigmoid函数将播放时间变换为一个在0-1之间的实数，以实现连续化。

- **视频完播率建模：**
  - **回归方法：** 用模型输出 p 直接逼近实际播放率 y（如 实际播放视频总长），使用标准交叉熵回归。

$$ss = y \cdot \text{o}p + (1\ y) \cdot \text{o}\,(1\ p)$$

  - **二元分类方法：** 人为定义一个完播硬阈值（如播放量 > 80 为正样本），将其转化为标准的二分类模型，预估“能完播”的概率。
  - **视频时长偏置修正：** 完播率天生对短视频有利、对长视频不公平。线上必须引入基准完播曲线函数 $f(\text{视频长度})$ 进行平滑调整，消除长视频的完播劣势：$p_{\text{e}} = \frac{\text{预估完播率}}{f(\text{视频长度})}$。



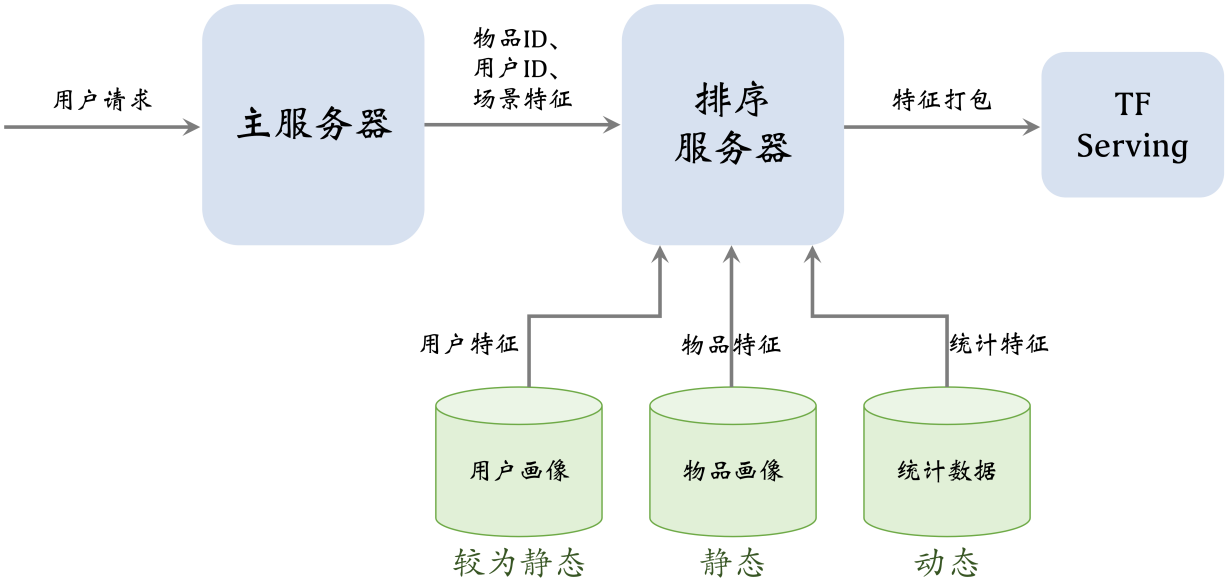
5. 排序模型特征与数据服务

特征分类系统

1. **用户画像（User Profile）：** 用户 ID Embedding、人口统计学属性（性别、年龄）、账号活跃状态及长期兴趣标签（关键词、品牌） 。
2. **物品画像（Item Profile）：** 笔记 ID Embedding、发布时间/年龄、地理位置编码（GeoHash）、文本元数据（标题、字数、清晰度）及美学/内容质量分 。
3. **用户统计特征：** 用户在近30天、7天、1天及1小时内的全盘行为统计，并按笔记展现形式（图文/视频）或内容类目（美妆、美食等）进行细粒度分桶 。
4. **笔记统计特征：** 笔记自身的分时段消费指标（曝光量、点击率、点赞率等），以及作者维度的全局质量特征（粉丝数、历史发布量） 。
5. **场景特征（Context）：** 用户实时定位 GeoHash、当前时间戳分段、是否周末/节假日、手机软硬件设备型号 。

特征工程与数据流工程

- **非线性特征变换：**
  - 高维离散特征: 直接做低维稠密 Embedding；
  - 连续特征（如年龄、视频长度）: 采用分桶离散化；
  - 高维计数特征（如曝光数、点赞数）: 执行 $o(1+)$ 平滑变换 。
- **特征覆盖率提升：** 诸如年龄、地理定位等特征由于隐私或未填写原因覆盖率远低于 100%，提升特征覆盖率是优化精排模型准确性的关键工程方向 。
- **高效数据服务架构：** 数据服务系统将特征划分为静态与动态。主服务器接收用户请求后派发 ID，排序服务器从高性能特征缓存区实时提取大规模静态画像与动态统计数据并进行特征组装，最终打包推送给 TF Serving 进行高并发模型推理 。



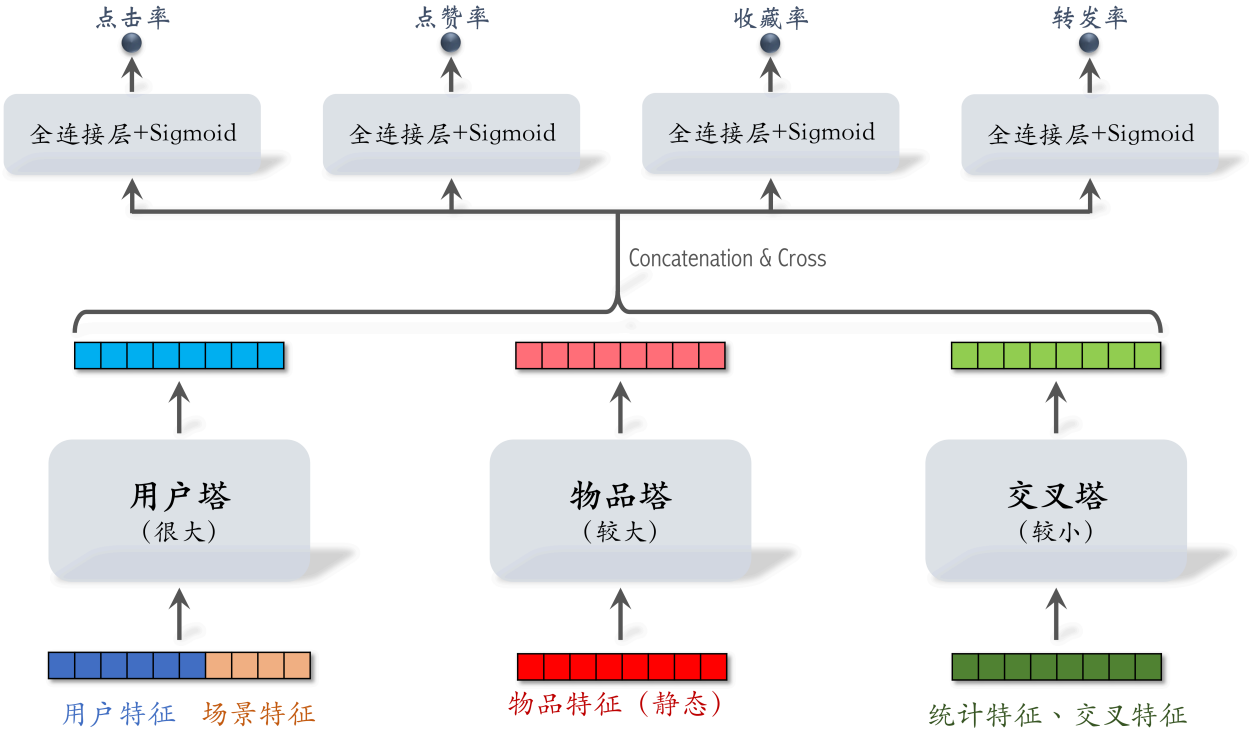
6. 粗排三塔模型 (COLD 架构)

粗排 vs 精排模型对比

- **粗排模型：** 面向几千篇候选笔记，单次推理代价必须极小，预估准确度要求相对折中 。所以，一般采取后期融合策略。
- **精排模型：** 面向几百篇候选笔记，单次推理特征交叉极其繁重，算力开销巨大，但追求极致的准确率 。所以，一般采取前期融合策略。
- **双塔模型缺陷：** 传统双塔将用户和物品解耦，线上虽只需推理一次用户塔（物品向量直接查表），算力极省，但由于缺乏底层特征交叉，预估准确度显著低于精排，难以作为高性能粗排 。

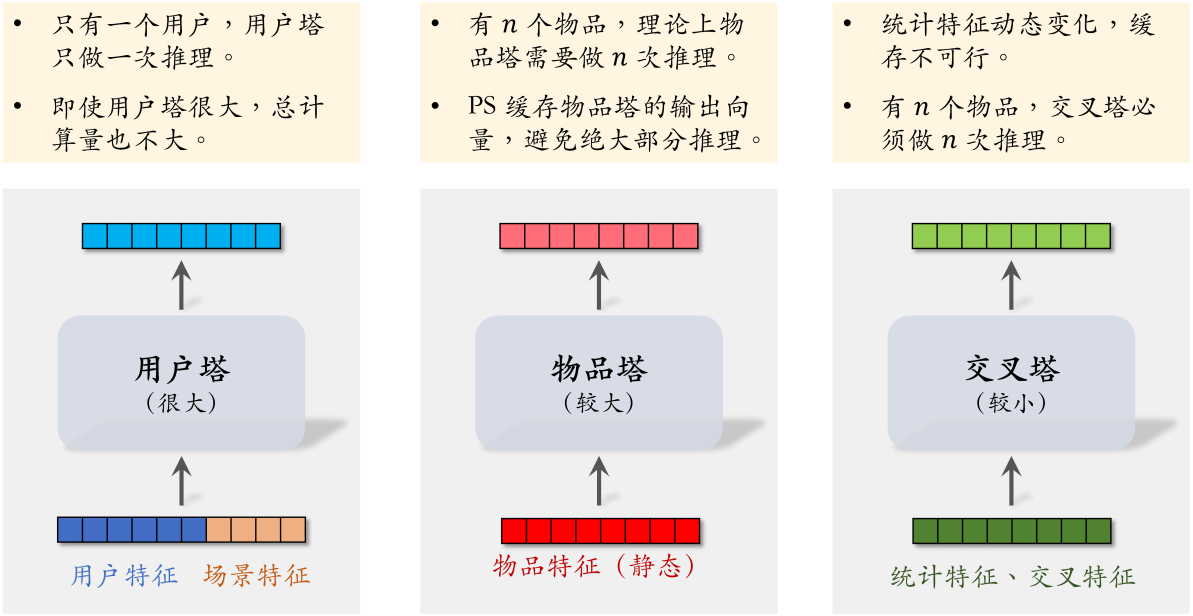
COLD 三塔模型设计与缓存机制

- **阿里 COLD 三塔结构：** 为了在算力与交叉特征间取得平衡，模型被拆分为：**用户塔**（包含用户与场景特征，模型庞大）、**物品塔**（仅包含静态物品特征，模型较大）和**交叉塔**（包含动态统计特征与轻量级交叉特征，模型精简） 。

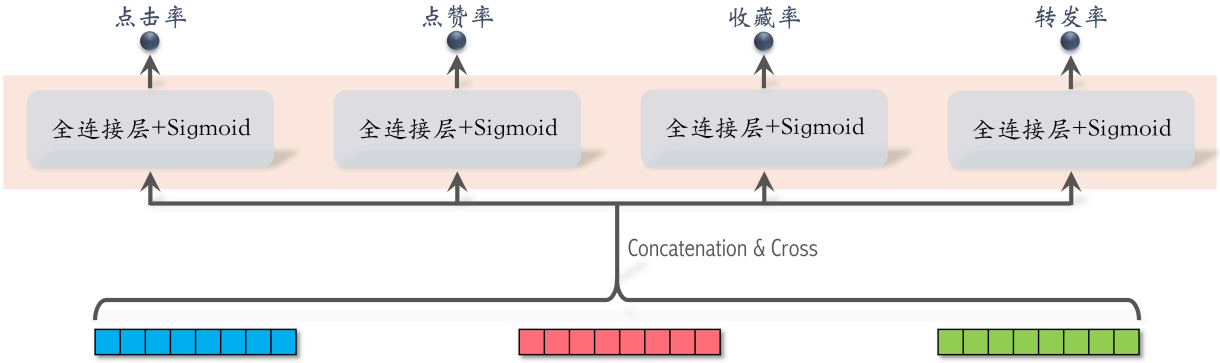


- **极速推理分工与优化：**

- **用户塔：** 单次请求中只有 1 个用户，用户塔在线上只需执行 1 次网络推理，计算量不随物品数膨胀 。
- **物品塔：** 输入完全依赖静态物品特征。因此，其网络输出向量可直接静态缓存在参数服务器（PS）中，线上直接命中缓存查表，免去 99% 的网络推理开销 。



- **交叉塔与上层模型：** 动态统计特征实时变化无法缓存，必须针对 n 个候选物品进行 n 次轻量级交叉特征推理，粗排绝大部分线上算力开销被成功压缩并集中于此 。



- 有 n 个物品，模型上层需要做 n 次推理。
- 粗排推理的大部分计算量在模型上层。

💡 Key Takeaways

多目标排序模型是工业级推荐系统中决定用户核心转化率与停留时长的精细化控制中枢，它通过将离散、连续等高维异构特征相结合，并利用 MMoE 等参数隔离架构实现点击、点赞、收藏等多个业务目标的联合建模 。在实际的工业工程落地中，系统往往需要通过负样本降采样预估值校准、视频时长加权交叉熵、完播率偏置调整等一系列严谨的数理策略，来确保线上推理分数的物理含义客观准确 。为了在有限的线上耗时约束内发挥深度模型的威力，现代架构演化出了粗精排分级漏斗机制，并借助如 COLD 三塔模型的静态物品向量缓存等创新的工程解耦手段，完美在算法精度与计算时效之间找到了极致平衡 。

IV. 特征交叉

1. 因子分解机 (Factorization Machines)

线性模型与二阶交叉特征

- **基础线性模型：** 特征单独加权求和，缺乏特征之间的乘积交叉组合。

$$p = + \ _{i=1}^{d} w_{ii}$$

  该模型在 d 个特征下面临的参数量为 $d+1$ 个。
- **二阶交叉特征模型：** 在线性模型基础上显式加入了任意两个特征的乘积交叉项，用以捕获特征间的关联性。

$$p = + \ _{i=1}^{d} w_{ii} + \ _{i=1}^{d} \ _{j=i+1}^{d} u_{ijij}$$

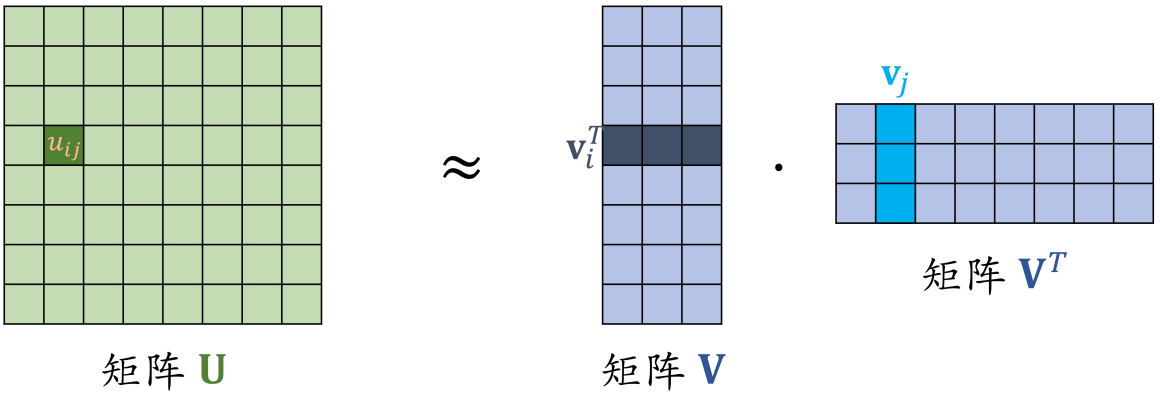
  - 尽管这种组合的表达能力比单纯线性模型更强，但导致二阶权重的总参数量急剧膨胀至 (d)。

FM 矩阵分解低维近似

- **低维隐向量内积替代：** FM 的核心创新点在于对庞大的系数矩阵进行低维分解，用两个低维稠密向量的内积来近似代替高维独立的交叉系数，即满足 $u_{ij} \ _{Tij}$。
- **参数量复杂度大幅优化：** 通过将每个特征映射到一个维度为 k $(k \ d)$ 的隐向量 $_i$，成功将原本 (d) 的二阶参数复杂度直线降低到了 (kd)。
- **通用场景替代：** 最终形成的 FM 预估公式为：

$$p = + \ _{i=1}^{d} w_{ii} + \ _{i=1}^{d} \ _{j=i+1}^{d} (_{Tij})_{ij}$$

  作为线性模型的优秀替代品，凡是能应用线性回归或逻辑回归的推荐排序和预测场景，都可以切换成表达力更强的 FM 模型。

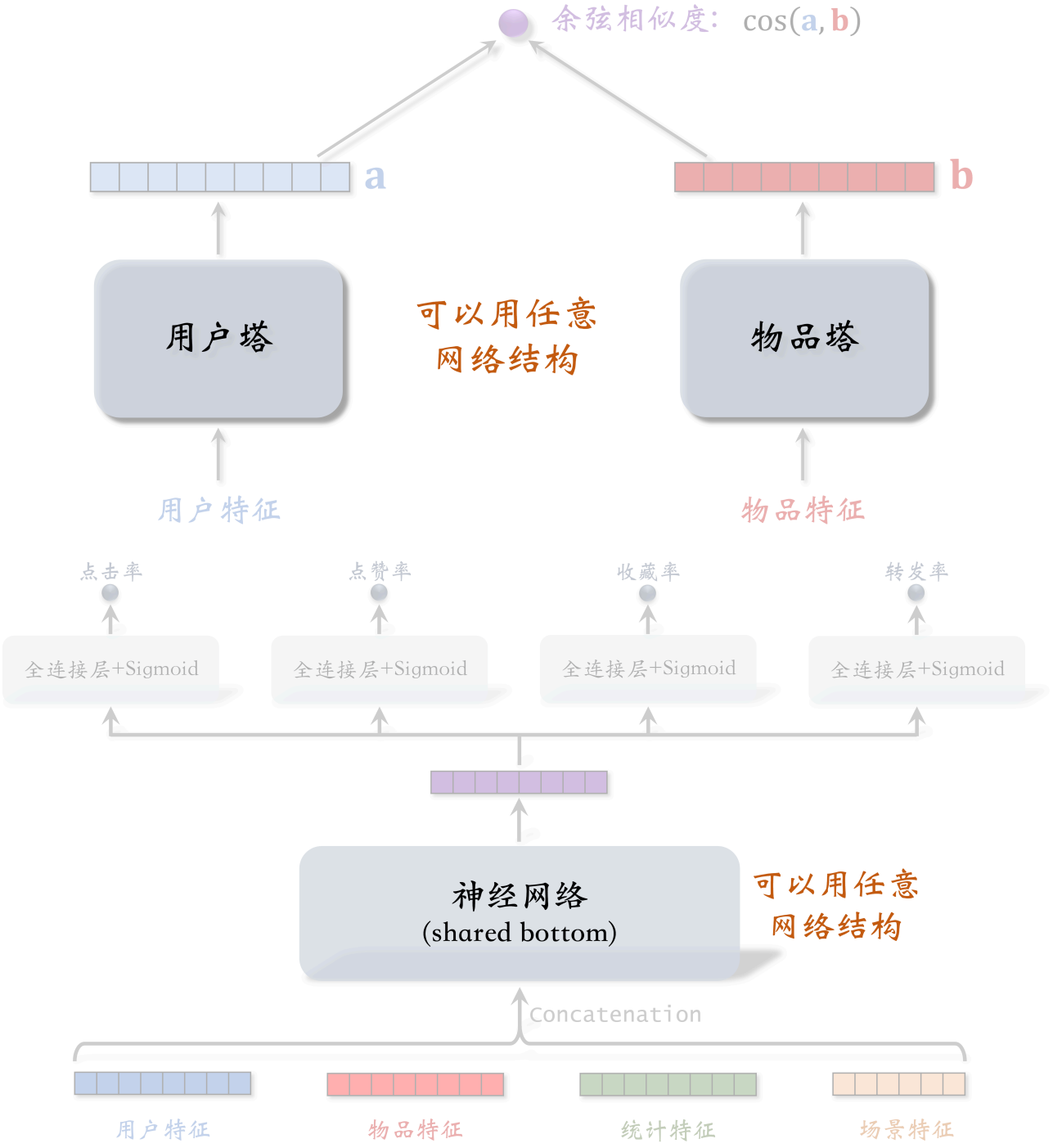


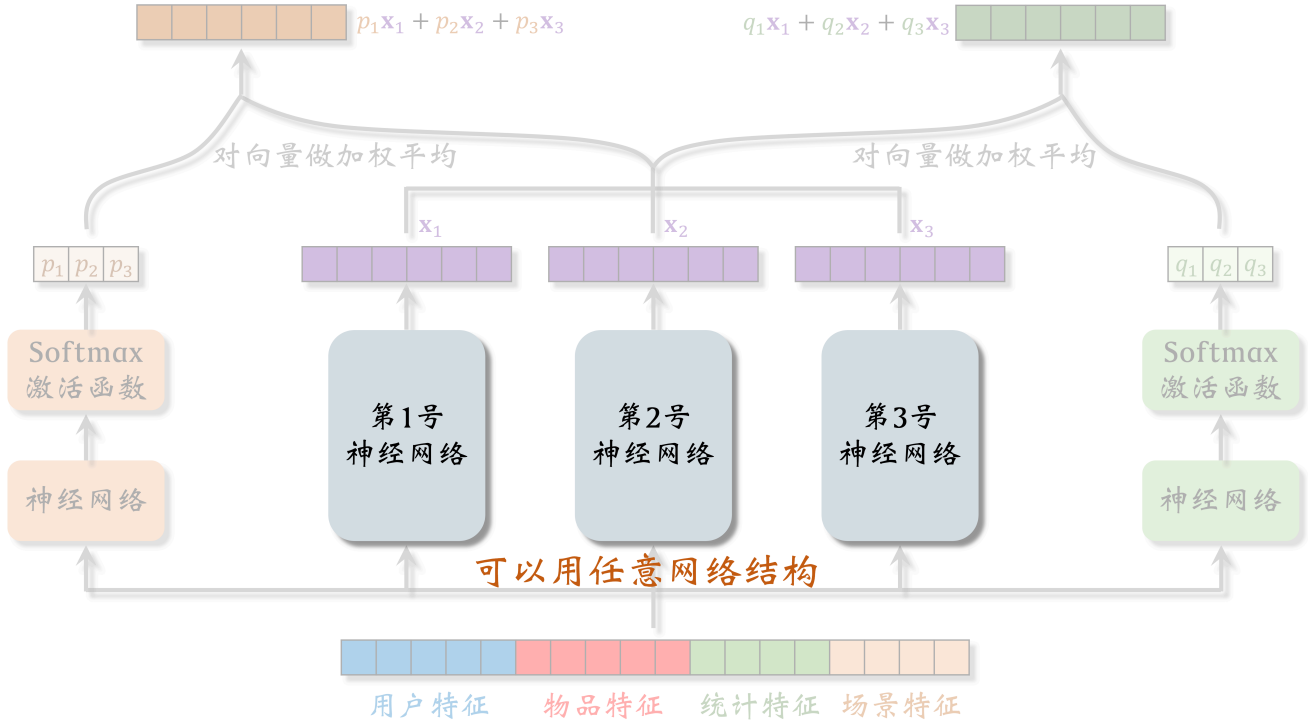
2. 深度交叉网络 (Deep & Cross Network)与多任务架构

多任务排序基础

- **底层共享网络模型 (Shared-Bottom)：** 用户特征、物品特征、统计特征和场景特征在底层拼接（Concatenation）后，统一送入底层共享的深度神经网络。
- **多分支并行输出塔：** 在网络顶部针对不同的业务指标独立搭建“全连接层 + Sigmoid”的任务分支，并行预估点击率、点赞率、收藏率和转发率。
- **示例：**

双塔模型

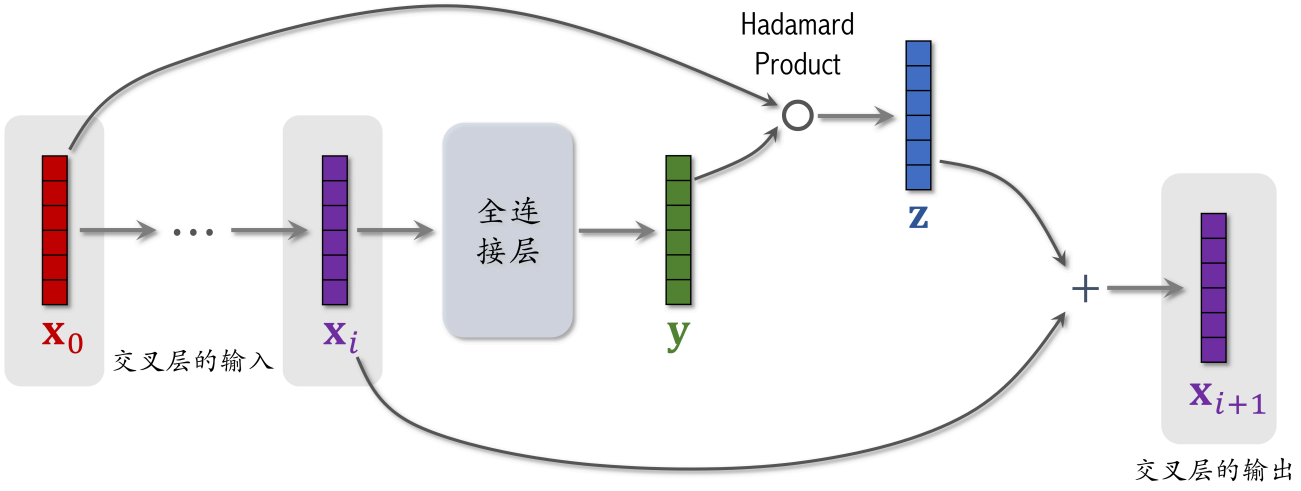

余弦相似度: cos(a, b)
a
b
用户塔
可以用任意
网络结构
物品塔
用户特征
物品特征
点击率
点赞率
收藏率
转发率
全连接层+Sigmoid
全连接层+Sigmoid
全连接层+Sigmoid
全连接层+Sigmoid
神经网络
(shared bottom)
可以用任意
网络结构
Concatenation
用户特征
物品特征
统计特征
场景特征



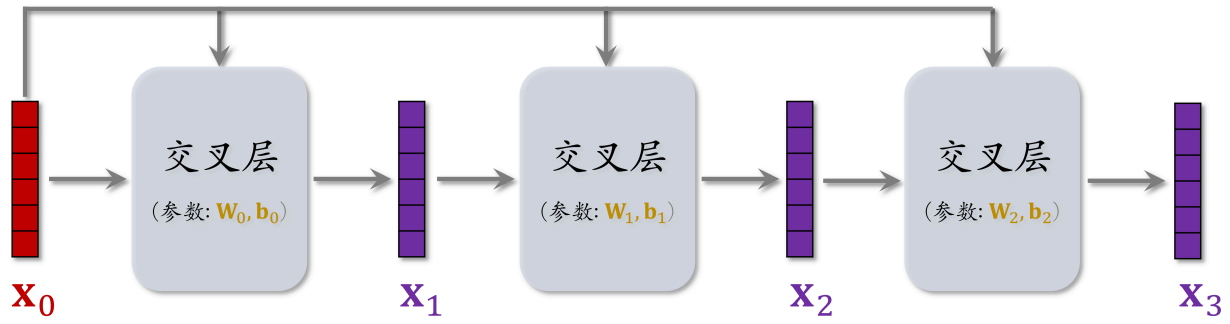
DCN 与 Cross Network V2 结构

- **网络阶数演进：** 传统的全连接层无法高效、显式地捕获高阶特征交叉，DCN 的提出正是为了解决该瓶颈。
- **Cross Layer 迭代公式：** DCN V2 通过交叉层让输入向量与当前层向量执行显式的哈达玛乘积（Hadamard Product）并引入残差连接。其单层核心迭代逻辑为：

$$_{+1} = _0 \left(+ \right) +$$



通过多层 Cross Layer 的连续堆叠，网络能够自动形成更高阶数的特征交叉特征。



- **双网络并行拼接：** 完整的 DCN 模型将负责显式高阶特征交叉的 **交叉网络 (Cross Network)** 与擅长隐式非线性拟合的 **全连接网络 (Deep Network)** 进行并行拼接组合，全方位融合两者的特征表征空间。

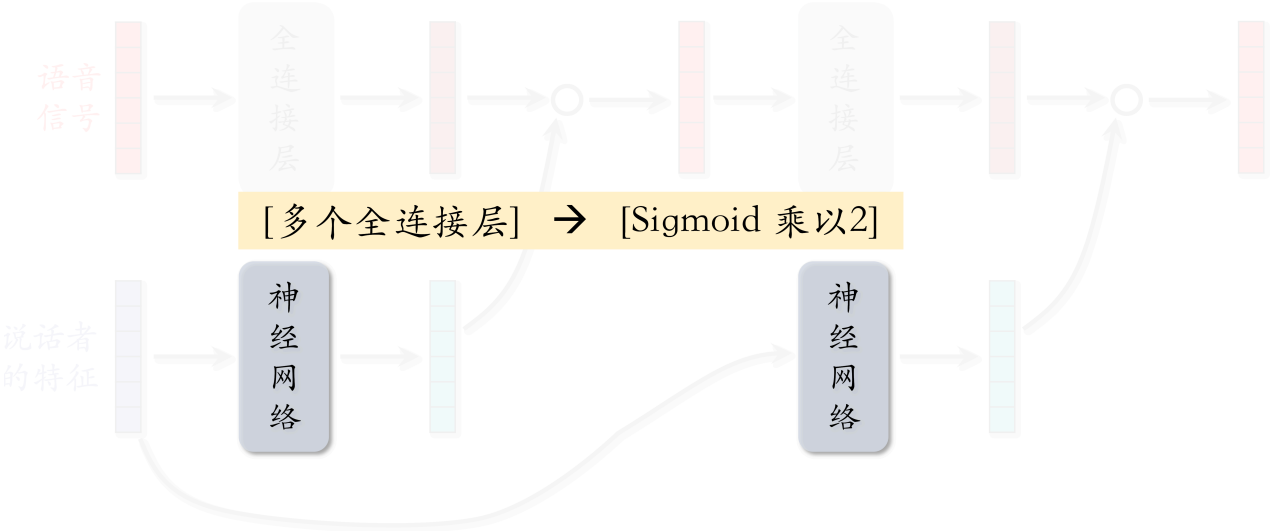
3. LHUC网络与PPNet

LHUC 的工业起源

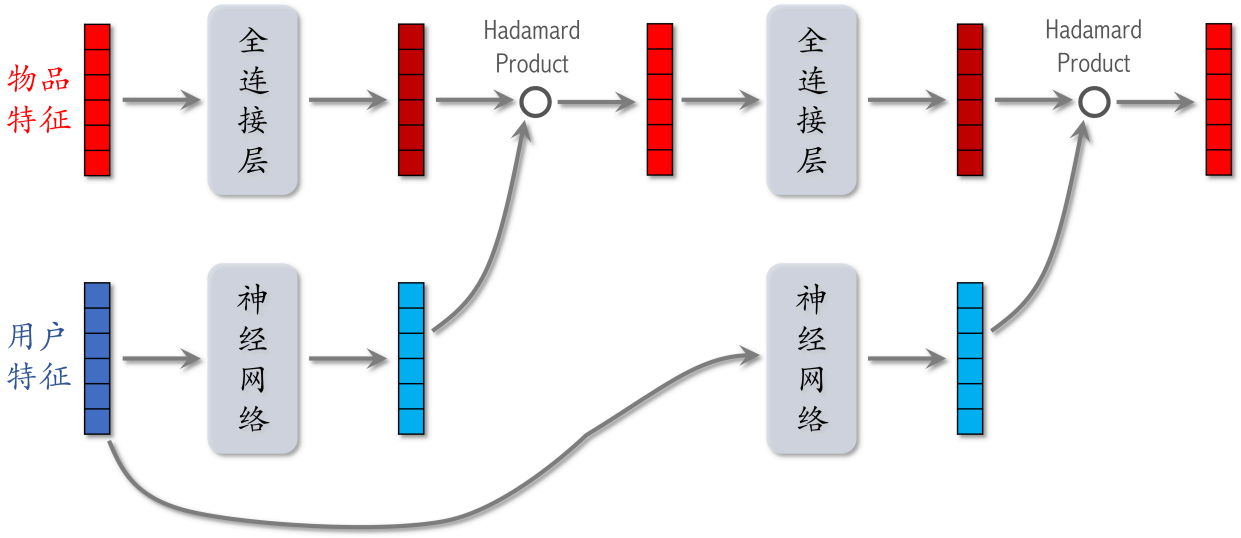
- **技术演进：** LHUC (Learning Hidden Unit Contributions) 机制最早起源于语音识别领域，通过引入说话者特征来动态调整主模型内部隐层神经元激活值的贡献度。
- **快手 PPNet 落地：** 快手将这一异构特征加权思想成功迁移并应用在推荐系统的精排排序阶段，构建了具备万亿参数规模的强个性化排序模型 PPNet。

推荐系统中的个性化运行机制

- **动态控制门控：** 在主网络的每个全连接层旁边，将当前的个性化偏置特征（如推荐系统中的用户特征、物品核心特征）输入到一个专属的浅层控制神经网络中。
- **两倍 Sigmoid 激活缩放：** 偏置特征经过控制网络计算后，通过 `Sigmoid` 激活函数并将输出结果直接**乘以 2**，以此将个性化缩放系数的范围限定在 0 之间（默认值为 1 代表保持原样，不放大也不抑制）。



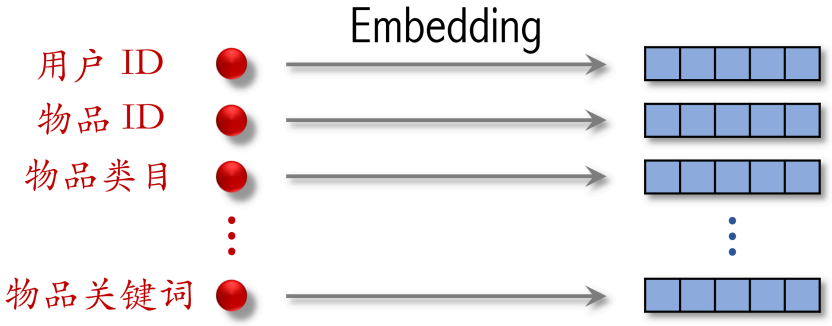
- **哈达玛乘积融合：** 利用该动态缩放系数与标准主网络中全连接层（FC Layer）的输出向量执行逐元素对应相乘（Hadamard Product），完成特征层面的深度个性化微调。



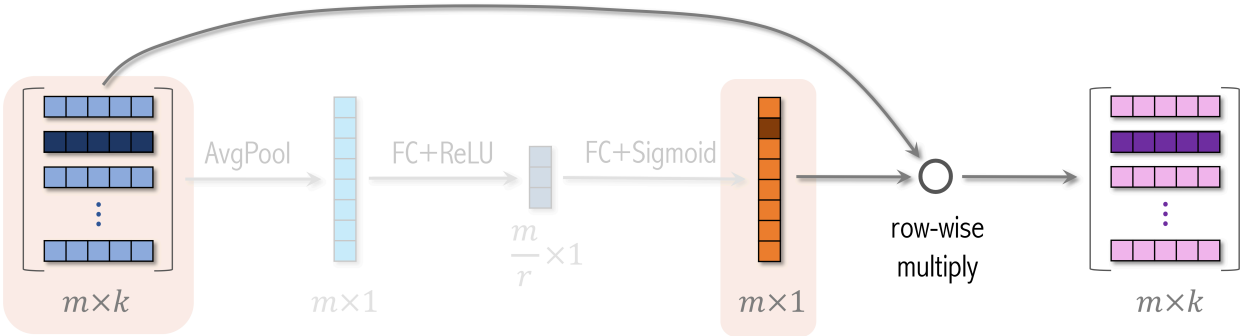
4. SENet与双线性特征交叉 (FiBiNet)

SENet 离散特征加权

- **Field-wise 权重自适应学习：** 传统排序模型中不同 Field（如用户 ID、物品 ID、物品类目、关键词）的离散 Embedding 被无差别使用，而 SENet 旨在根据特征上下文动态学习每个 Field 的重要性权重。
- **经典三步特征加工架构：**
  1. **Squeeze (压缩阶段)：** 对含有 个 Field、各 Field 维度为 k 的离散特征 Embedding 矩阵执行全局平均池化（AvgPool），将 k 的张量压缩为 1 的 Field 标量表征向量。

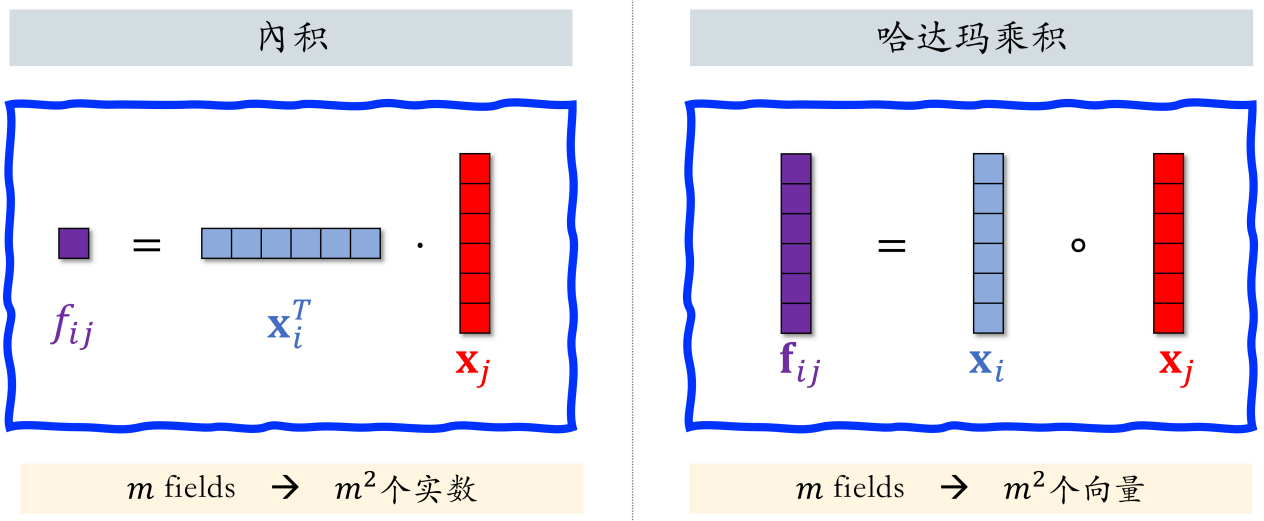


  2. **Excitation (激励阶段)：** 将压缩后的向量送入两层轻量级全连接网络（`FC + ReLU` → `FC + Sigmoid`）进行非线性映射，动态输出针对这 个 Field 的特征重要性权重向量 。(图中，常数为压缩比列。)
  3. **Reweight (重加权阶段)：** 将权重向量 与原始离散特征 Embedding 矩阵进行逐行向量相乘（Row-wise Multiply），完成 field-wise 维度的特征自适应增强。

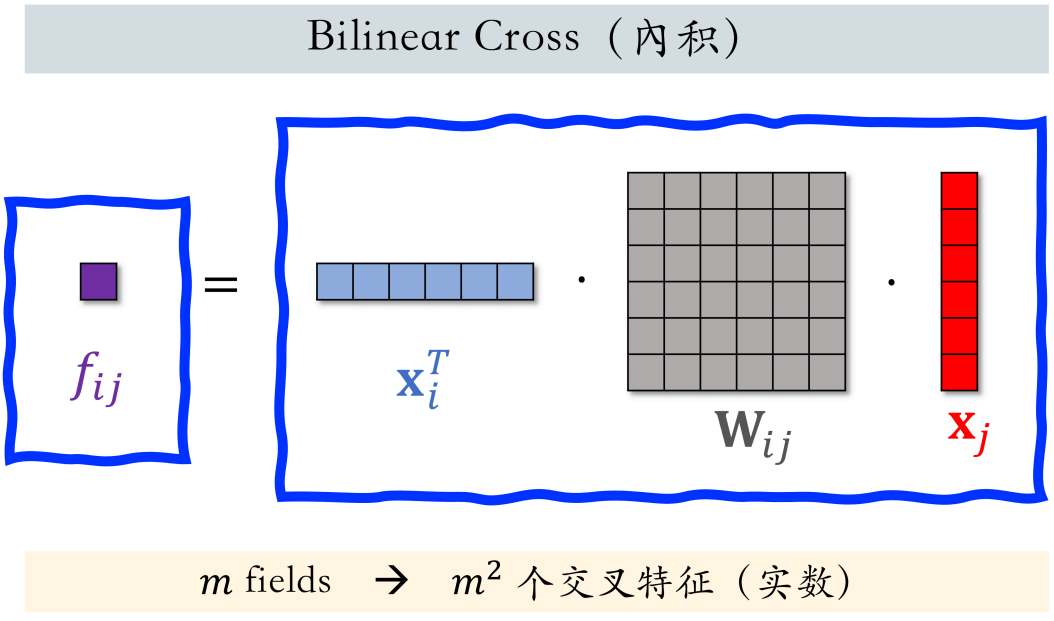


双线性特征交叉 (Bilinear Cross) 与 FiBiNet

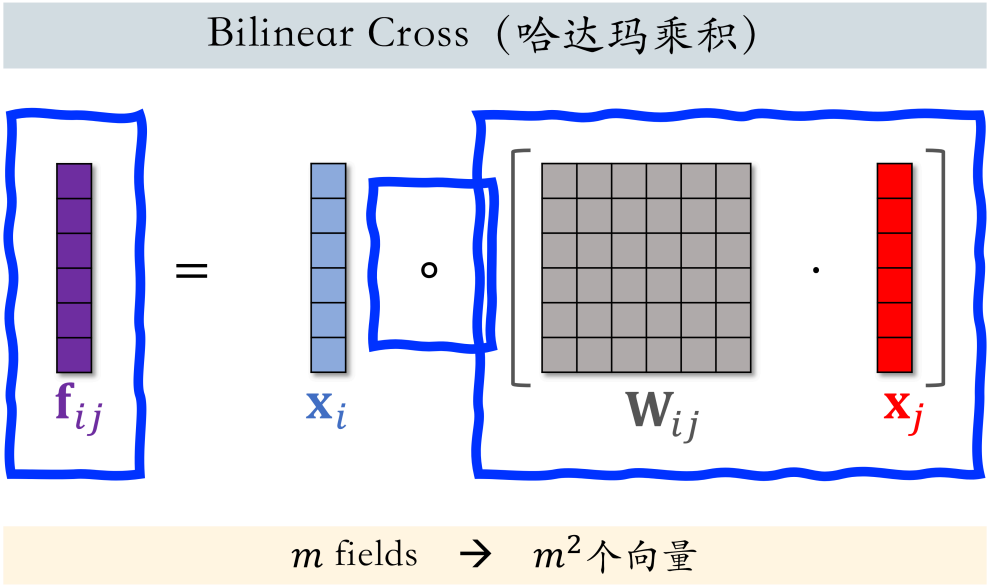
- **传统交互方式局限性：** 传统的 Field 间交互要么使用标量内积（无法保留多维空间向量信息），要么使用简单的哈达玛乘积（缺乏可训练的参数矩阵），限制了模型的交叉精度。



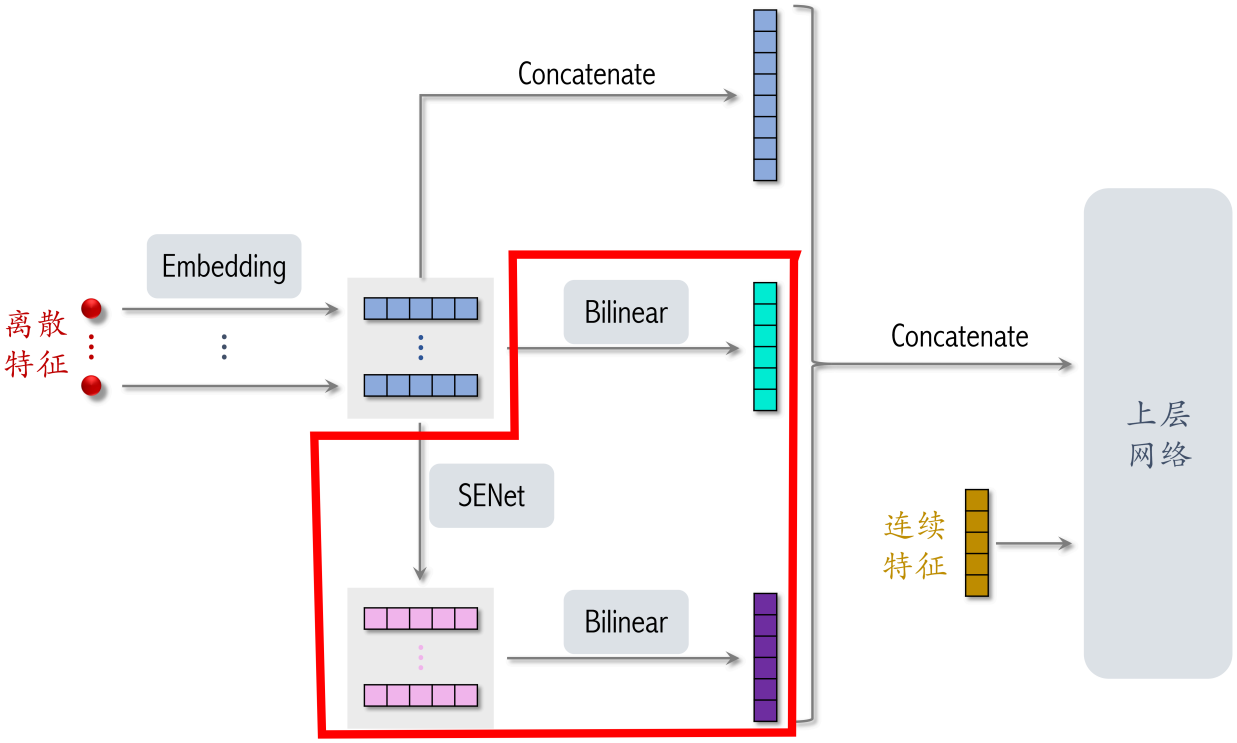
- **Bilinear Cross 的主流模式：**
  - **内积模式：** 在两个 Field 向量交互时引入可训练参数矩阵 $_{ij}$，使 个 Fields 在两两内积交叉后产生 个极具表征力的交叉特征实数。



  - **哈达玛乘积模式：** 通过形式如 $_i$ $_{ij}$ 的矩阵空间转换后再与 $_j$ 作逐元素相乘，最终在 Field 间生成 个高维特征向量。



- **FiBiNet 全局网络整合：** 经典的 FiBiNet 架构同时将“原始 Embedding 矩阵”与“经过 SENet 增强后的 Embedding 矩阵”分别送入双线性交叉（Bilinear Interaction）层进行交叉组合，最后与连续特征一并拼接输送给上层网络执行点击率预估。



💡 Key Takeaways

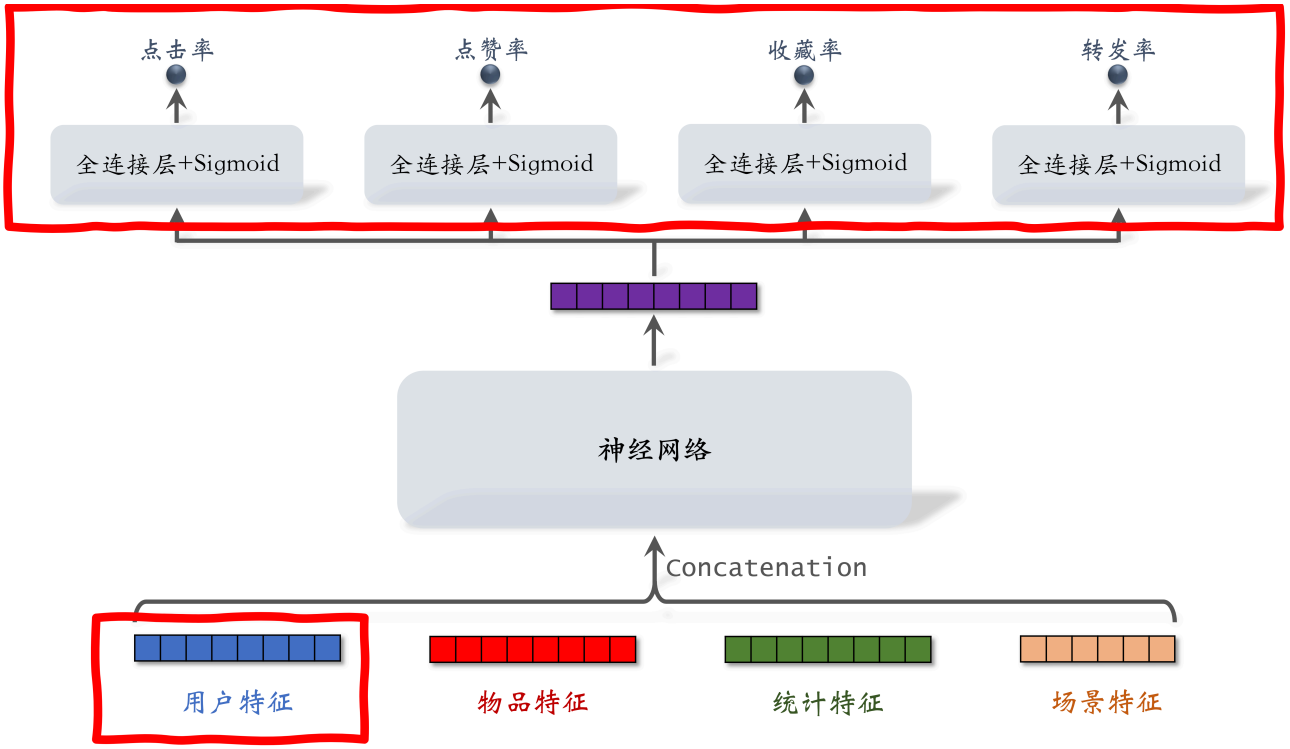
这篇 presentation 深入复盘了工业级推荐系统精排模型中围绕“显式特征高阶交叉”与“用户偏置动态加权”展开的核心演进脉络，展示了从经典 FM 模型的隐向量内积低维近似，到 DCN V2 利用显式哈达玛乘积构建任意高阶交叉层的技术飞跃。在此基础上，以快手 PPNet (LHUC 机制) 为代表的个性化微调架构通过引入特定的用户偏置特征，动态缩放主网络隐层神经元贡献度，大幅度强化了多任务学习下的个性化建模深度。最终，FiBiNet 模型通过完美融合 SENet 的 field-wise 离散特征通道注意力机制与多模态双线性特征交叉 (Bilinear Cross)，实现了特征重要度自适应识别与精细化语义交互的工业级工程落地。

V. 行为序列

1. LastN 特征与推荐系统基础架构

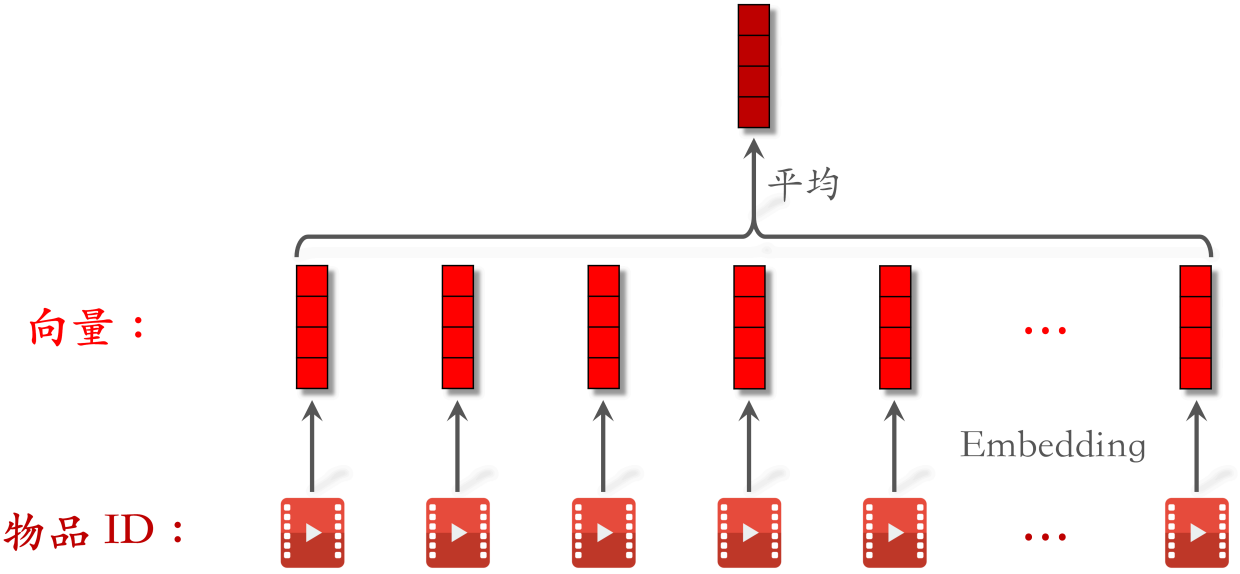
经典多目标排序模型

- **全连接神经网络基底：** 传统精排架构通过拼接（Concatenation）用户特征、物品特征、统计特征和场景特征，共同送入底层深度神经网络中。
- **多任务独立预估塔：** 网络顶层分化出多个并行的全连接层加 `Sigmoid` 激活函数，用于独立预估点击率、点赞率、收藏率、转发率等多维指标。



LastN 特征池化技术

- **基本定义：** LastN 指代用户近期发生交互（如点击、点赞等）的最近 n 个物品的 ID 序列。
- **简单平均池化：** 对这 n 个物品 ID 分别执行 Embedding 映射得到高维稠密向量，再通过直接求平均（Average Pooling）的方式，融合成一个代表用户交互历史的综合特征向量。

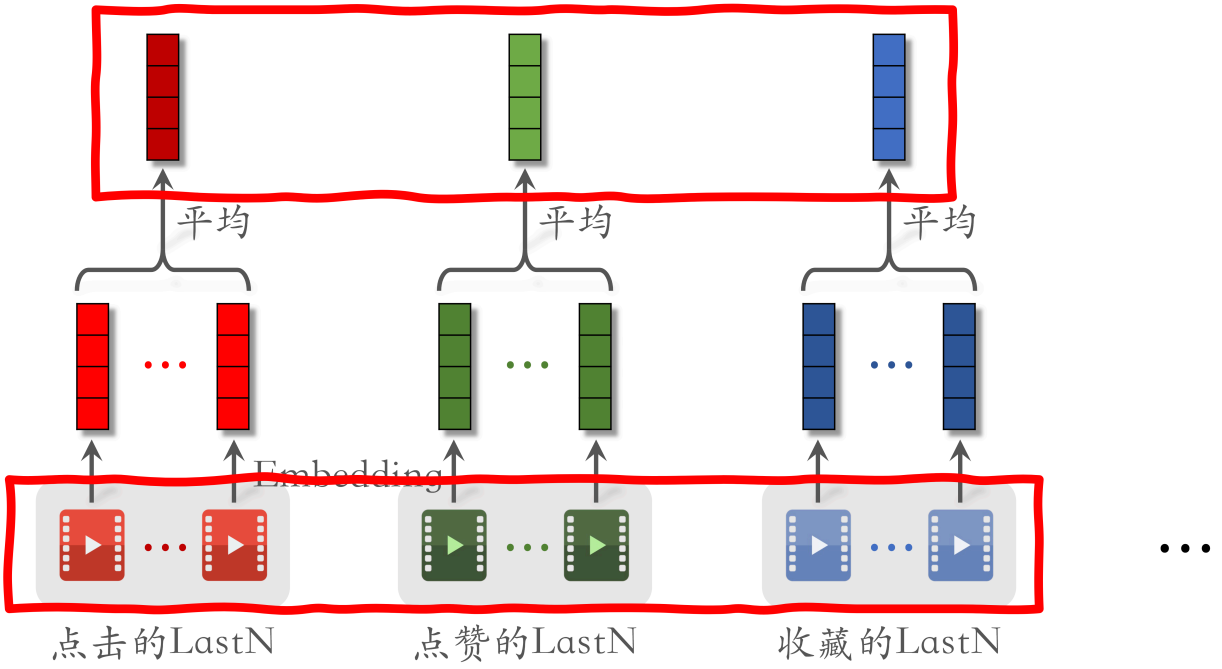


- **工业级应用局限性：** 这种依靠简单平均生成的特征属于用户自身的静态表征，虽然高度适用于双塔模型、粗排三塔模型和精排模型，但它无差别地对待所有历史行为，抹杀了用户在不同时刻、不同兴趣点之间的比重

差异。

小红书长序列落地实践

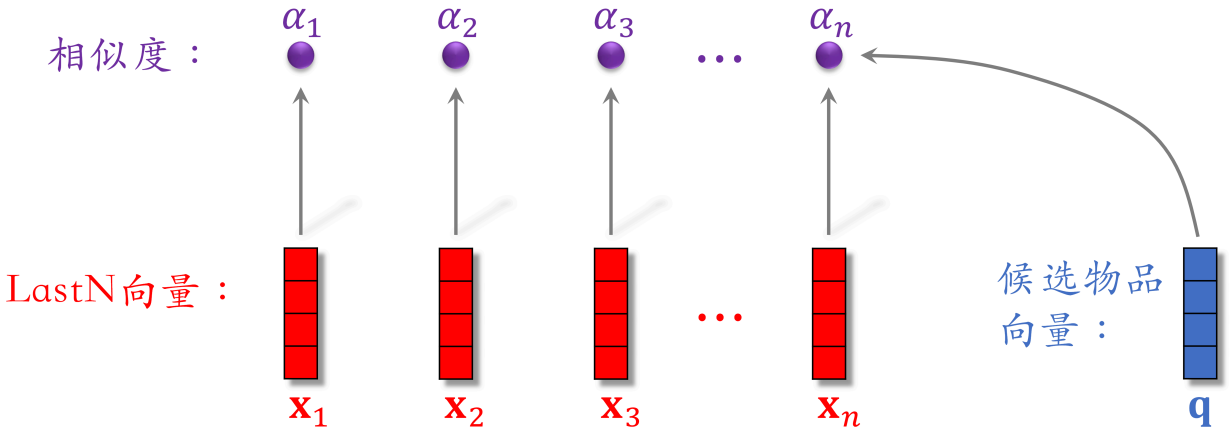
- **行为渠道细分：** 在具体的工业级落地中，系统并不会将所有交互混为一谈，而是针对点击、点赞、收藏等不同的行为渠道，分别建立独立的 LastN 行为特征序列。
- **并行网络池化：** 各渠道独立的交互序列各自执行 Embedding 编码与平均池化，生成多路独立的用户行为表征向量，并一同拼接入精排主网络中。



2. DIN 模型 (Deep Interest Network) 与注意力机制

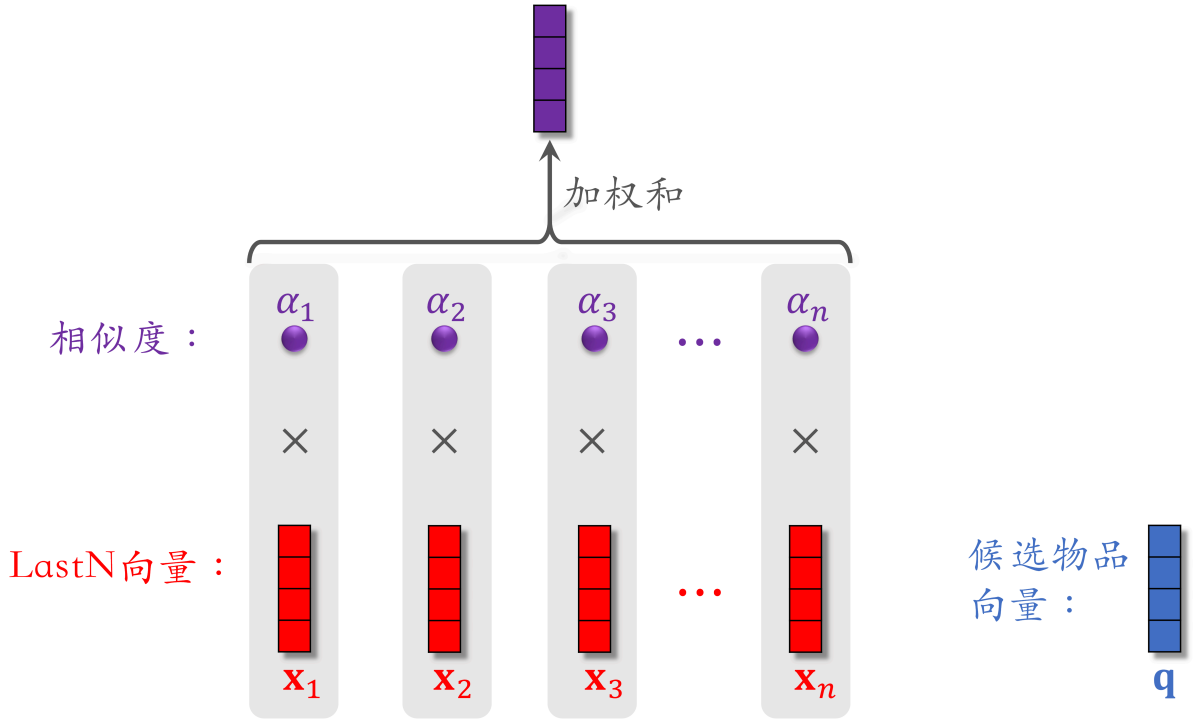
靶向注意力池化机制

- **核心升级逻辑：** 鉴于简单平均池化无法捕捉当前候选物品与历史行为的强关联，DIN 用基于注意力机制 (Attention Mechanism)的**加权平均**彻底取代了简单平均。
- **动态相关性权重：** 计算过程中，以当前需要预估的**候选物品向量 作为 Query**，用户历史的 **LastN 物品向量** $_i$ **分别作为 Key 和 Value**。



- **计算公式：** 通过注意力单头网络计算出候选物品与各个历史物品之间的相似度，将其作为注意力权重 $_i$，最后对历史物品向量执行加权求和，输出高度关联当前上下文的用户兴趣特征向量：

$$\text{erere} = \ {}_{i=1}^{n} {}^{ii}$$



模型适用场景边界

- **精排模型专属：** 注意力机制由于在计算权重时必须深度依赖当前的候选物品信息（Query），因而无法提前在隔离的用户塔中独立计算，这也导致其**无法应用于召回双塔模型和粗排三塔模型**中。
- **性能与序列长度冲突：** DIN 注意力层的计算量与用户行为序列长度 n 呈现强正比例关系（oe n）。为了避免线上推理严重超时，工业界一般只能保留最近几百个物品，而这往往会造成系统过度关注用户短期兴趣，从而产生对长期兴趣的严重遗忘。

3. SIM 模型 (Search-based Interest Model) 与长序列演进

双阶段“查找 + 注意力”架构

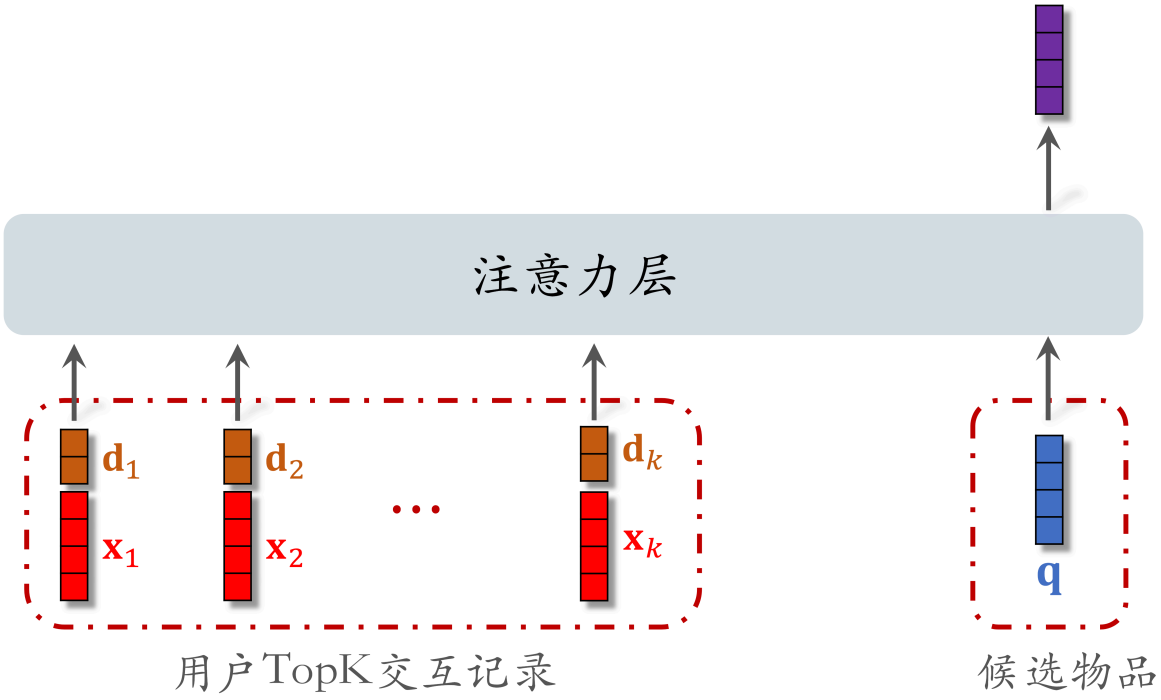
- **工业解决目标：** SIM 模型的提出旨在彻底解决 DIN 无法处理超长序列的算力瓶颈，在保证计算量不爆炸的前提下，成功将用户全生命周期行为序列 n 的支持规模推向数千级别。
- **核心破局思路：** 如果某个历史行为物品与当前候选物品差异极大，其相似度权重本就无限趋近于零。SIM 借此在进入注意力机制前，先通过第一步的快速检索，彻底剔除无关行为，使进入核心计算的序列长度大幅缩减为 k (k n)。

阶段一：多样化查找策略（Search Stage）

1. **Hard Search（硬查找）：** 直接根据当前候选物品的类目（Category），在用户全量历史序列中强行过滤并仅保留类目完全相同的历史物品。该方式极为简单、计算飞快且完全无需训练。
2. **Soft Search（软查找）：** 将所有物品编码为稠密 Embedding 向量，以候选物品向量作为 Query，在超长历史序列中执行快速的 k 近邻（KNN）检索，保留最相似的 k 个物品向量。该方法具备更强的语义捕获效果，但工程落地与代码实现较为复杂。

阶段二：融合时间特征的注意力机制（Attention Stage）

- **时间遗忘模型：** 在第二阶段对筛选出的 Top-k 行为向量执行注意力加权时，由于长序列涵盖的时间跨度极大（时间越久远，重要性越低），必须引入时间衰减增量。
- **时间编码拼接：** 计算当前距用户与该历史物品交互时刻的时差 ，对 进行离散化处理后训练专属的时间 Embedding，生成时间向量 。最终将物品 Embedding 向量 与时间 Embedding 向量 执行拼接（Concatenation），合并输出为 投入注意力层进行聚合表征。



核心选型结总结

- **长序列优于短序列：** 融入全生命周期长序列（长期兴趣）的表征能力在各大指标上均显著优于仅截取短期的近期交互特征。
- **加权优于简单平均：** 在排序精细化建模中，注意力机制 (Attention Mechanism) 捕捉动态靶向偏好的成效明显优于无差别平均。
- **软硬检索依基建而定：** Soft Search 与 Hard Search 的路线选择并不存在绝对的优劣之分，完全取决于业务场景及团队内部的工程基建成熟度。

💡 Key Takeaways

用户行为序列建模在推荐系统排序阶段经历了从静态平均到靶向动态感知、再到全生命周期长序列演进的跨越式发展。最初的 LastN 简单平均池化虽然对基础工程架构极其友好，却抹杀了用户多元兴趣的侧重点；而 DIN 模型通过引入目标注意力机制完美实现了动态权重划分，但其计算复杂度随行为长度线性增长的缺陷制约了对长期偏好的捕捉。最终，SIM 模型凭借双阶段“检索过滤 + 靶向注意力”的破局思路，配合时空重加权编码，成功在算力可控的边界内实现了万级超长序列的工业级落地。

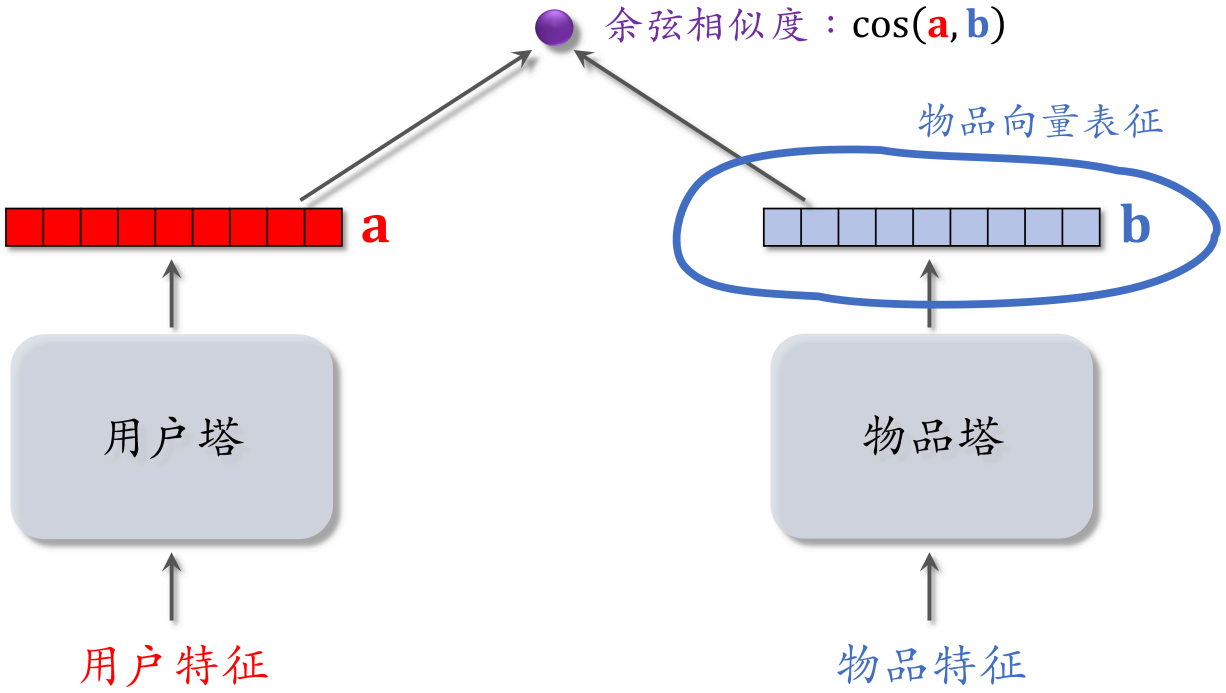
VI. 重排

1. 物品相似性度量与多模态表征

属性标签与基础向量表征

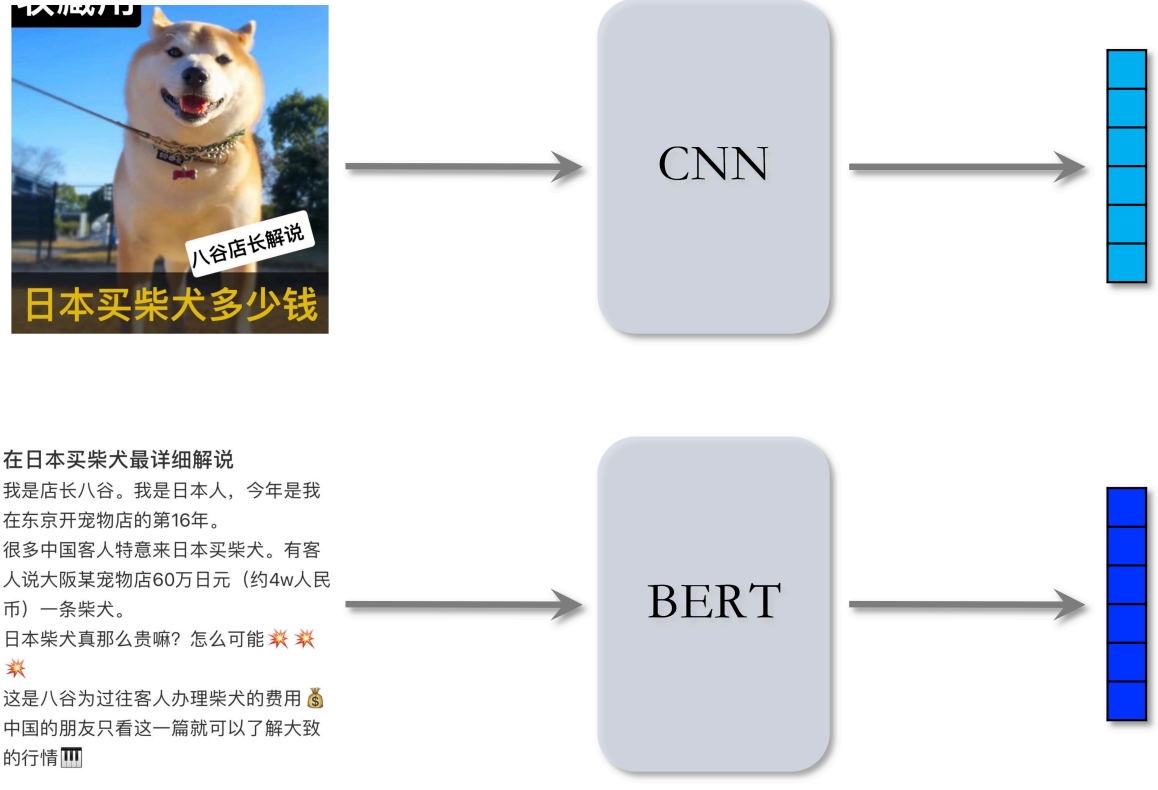
- **基于属性标签：** 通过比对一级类目、二级类目、品牌和关键词等显式标签来直接计算离散相似度。
- **双塔模型向量的局限性：** 召回或粗排双塔模型学习到的物品向量，主要为了匹配用户潜在偏好，在重排阶段直接拿来衡量两个物品之间的纯粹多样性相似度往往效果欠佳。

双塔模型的物品向量表征



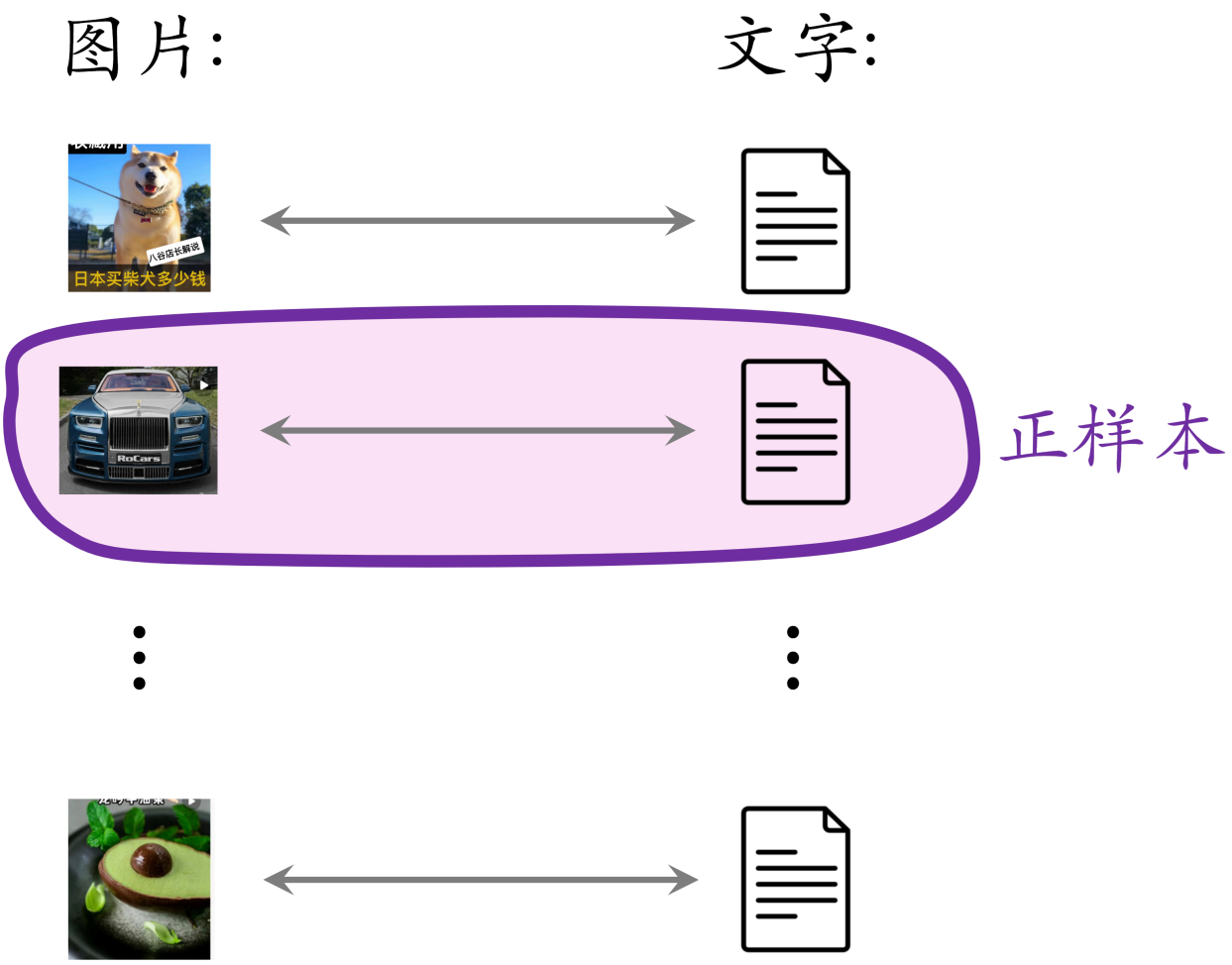
- **基于内容的向量表征：** 能够真实捕捉图文、音视频底层语义，是重排阶段多样性计算的首选。通过图像神经网络（如 CNN）提取视觉表征，或通过预训练语言模型（如 BERT）提取文本表征。但是，该方法受限于标注数据的规模，训练成本高昂。

基于图文内容的物品表征

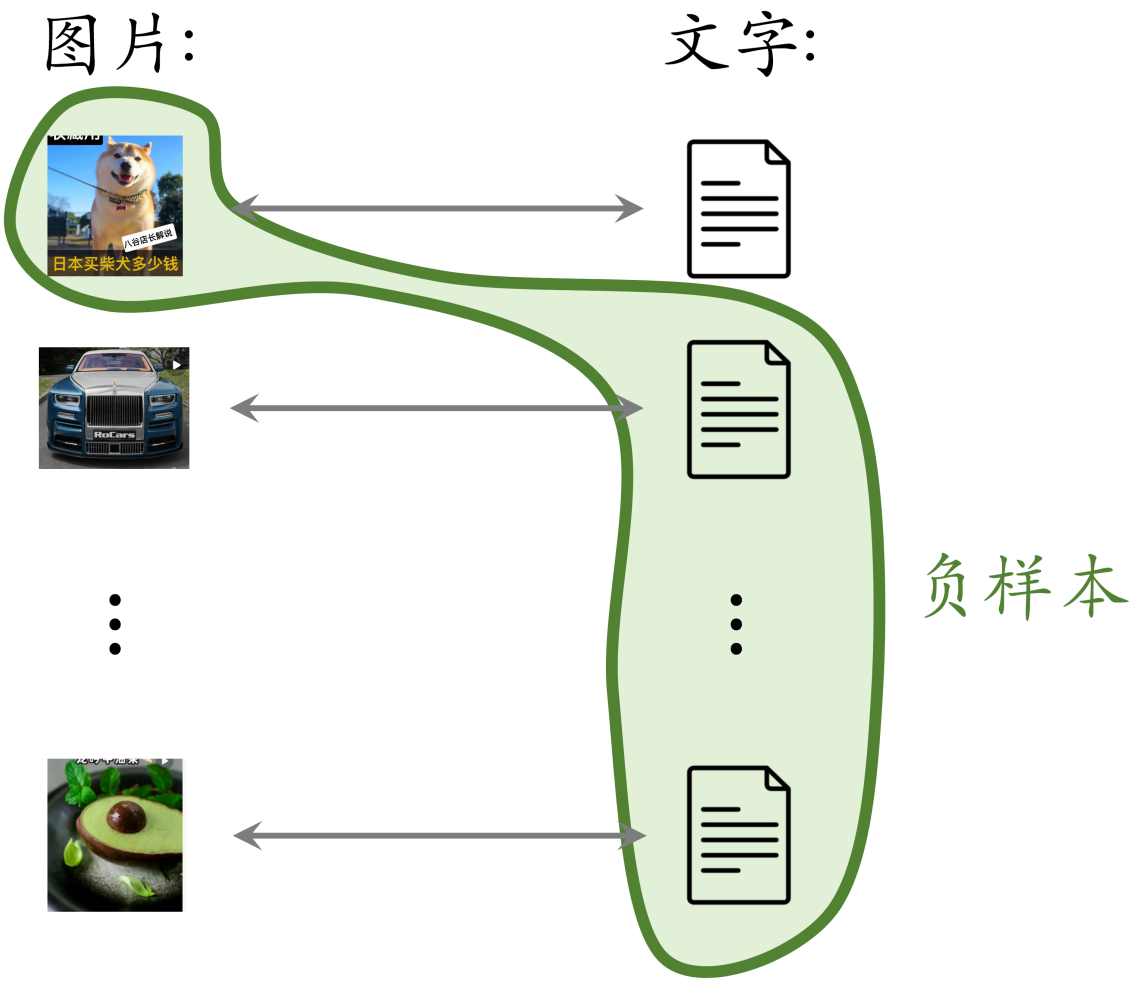


工业级 CLIP 预训练多模态表征

- **CLIP 核心思想：** 基于对比学习（Contrastive Learning）机制，对海量“图片-文本”二元组进行训练，预测图片和文本是否来自同一篇笔记。
- **免人工标注优势：** 小红书的笔记天然将图像与文本深度绑定，无需任何额外标注成本，即可形成大规模高质量的多模态联合特征空间。



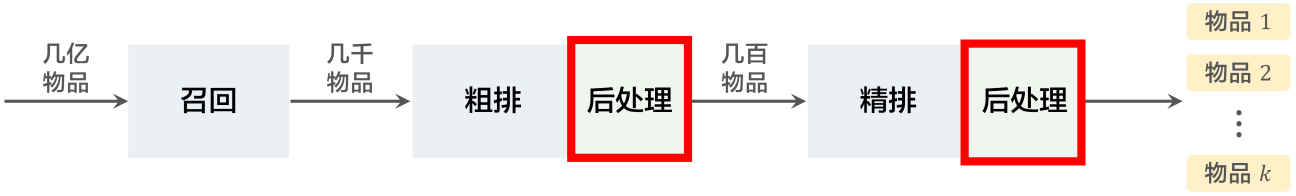
- **负样本构造机制：** 在一个包含 对正样本的训练 Batch 中，任意一张图片和非它配对的 1 条文本均组成负样本，整个 Batch 内共可派生出 (1) 对负样本用于高效对比学习训练。



2. 最大边际相关性 (MMR)算法

推荐系统重排链路定位

- **全链路瓶颈：** 粗排与精排阶段多采用逐个物品独立打分（Pointwise Scoring）的范式，模型输出的 `reward_i` 仅代表单品的绝对价值或用户对其的独立兴趣分。
- **重排（Rerank）的核心任务：** 从精排产生的 n 个高分候选物品中选出 k 个物品呈现在终端。目标既要保证候选集整体的精排总分（`reward`）最大化，也要保证集合内部物品具有足够的分散性与多样性，避免同质化内容扎堆曝光。



- 给定 n 个候选物品，排序模型打分 $\text{reward}_1, \cdots, \text{reward}_n$。
- 从 n 个候选物品中选出 k 个，既要它们的总分高，也需要它们有多样性。

标准 MMR 贪心机制与公式

- **计算公式：** MMR 属于一种典型的启发式贪心选择算法，每轮计算未选集合 中各物品的边际相关性分数（Marginal Relevance）：

$$_i = \cdot \text{rer}_i \ (1 \ ) \cdot \ _{j\mathcal{S}}(i\,j)$$

通过超参数 0 1 动态权衡物品本身的绝对精排价值（前项）与该物品相较于已选集合 $\mathcal{S}$ 的最大相似度惩罚（后项）。



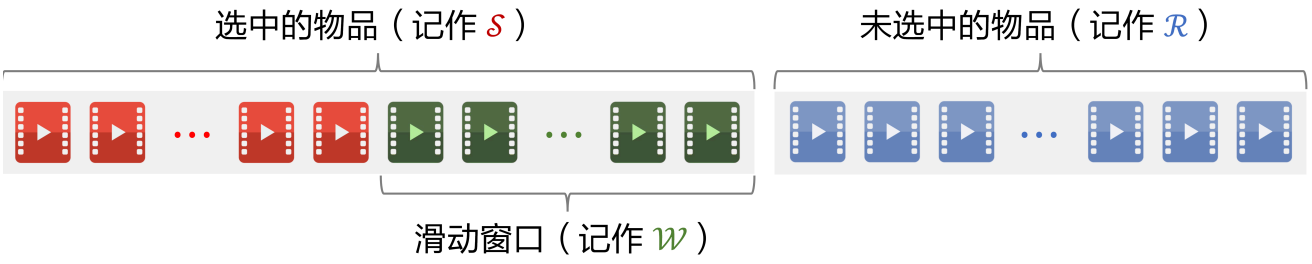
- **算法执行步骤：**
  1. 初始化已选集 $\mathcal{S} =$ ，候选全集 $= 1 \ n$。
  2. 首轮直接将绝对精排分数 `reward` 最高的单品移入已选集 $\mathcal{S}$。
  3. 执行 k 1 轮循环，每轮计算全量 $_i$，并将最高边际分数的物品 r_{ii} 从 剔除并加入 $\mathcal{S}$。

滑动窗口降噪机制

- **核心失效痛点：** 随着算法演进，已选集合 $\mathcal{S}$ 的规模日益庞大，集合中包含的相似物品覆盖度随之急剧攀升。在相似度值域为 0 1 的情况下，任何新物品跟 $\mathcal{S}$ 中某件陈旧物品的 $(i\,j)$ 几乎都趋近于 1，导致 MMR 的多样性惩罚项彻底失效退化。
- **窗口裁剪（Sliding Window）：** 引入大小为 的滑动窗口 （例如仅保留最近曝光的 10 个物品）替代整个大集合 $\mathcal{S}$ 参与惩罚计算：

$$\text{r}_i \cdot \text{rer}_i \ (1 \ 0) \cdot \ _j(i\,j)$$

该操作有效规避了极长时空跨度下的长尾累积干扰，更贴近用户的即时视觉疲劳阈值。



3. 几何体积度量与行列式点过程 (DPP) 算法

超平行体体积与多样性映射

- **几何物理学意义：** 给定 k 个物品，将其内容表征映射为 d 维空间中的单位向量 $_1 \ _k \ ^d$。这组向量可在高维空间中唯一确定一个 k 维超平行体（Parallelotope）。
- **体积与多样性的单调映射：**
  - **完美正交（多样性极佳）：** 当所有向量两两正交时，高维超平行体呈标准正方体，体积取得最大值 $= 1$。
  - **线性相关（多样性极差）：** 当向量完全线性相关或落在低维子空间内，高维体发生塌缩，体积取得最小值 $= 0$。
- **行列式转换公式：** 将这 k 个向量作为列构建矩阵 ^{dk}，其对应的 Gram 矩阵的行列式值与其生成的超平行体体积具有二次幂等价关系：

$$\mathrm{e}(^T) = ((_1 \ _k))$$

因此，可以使用子矩阵的行列式作为度量集合元素多样性的代数利器。

DPP 组合优化建模

- **全局协同优化目标：** Hulu 将传统的统计机器学习方法 DPP 引入重排场景，通过将绝对单品价值与空间几何体积统一融合，把重排升级为一个完备的数学最优化问题：

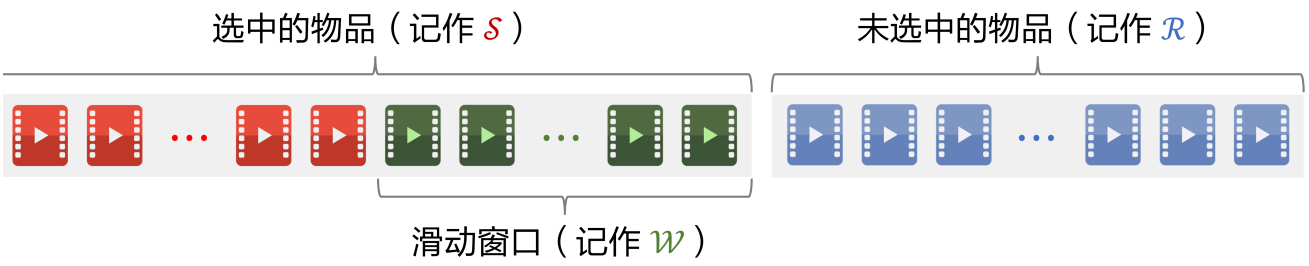
$$\mathrm{r}_{\mathcal{SS}=k} \cdot \ _{i\mathcal{S}} \mathrm{rer}_i + (1 \ ) \cdot \mathrm{o}\,\mathrm{e}(_{T\mathcal{S}})$$

- **计算效率跃升（Hulu 快速算法）：**
  - **暴力贪心代价：** 逐一求解 $\mathrm{e}(_{\mathcal{S}i})$ 矩阵行列式的原生代价为 $(\mathcal{S}^3)$，直接暴力的总时间复杂度高达 (nk)，无法满足线上高并发的低时延要求。注：含求解A的总时间复杂度为$(nd + nk)$.
  - **Cholesky 增量分解：** Hulu 团队提出对核矩阵进行 Cholesky 增量分解 $_{\mathcal{S}} = LL^T$（L 为下三角矩阵），利用下三角对角线乘积快速求出矩阵行列式：$\mathrm{e}(_{\mathcal{S}}) = \ _{ii}$。通过前一步分解矩阵的高效递推复用，成功将重排求解的时间复杂度彻底降至 $(nd + nk)$ 的工业可用量级。

DPP 下的滑动窗口扩展

- **长序列行列式坍缩：** 在长序列重排中，已选集 $\mathcal{S}$ 不断扩张会不可避免地导致其中包含的相似向量愈发趋于线性相关，进而触发行列式值 $\mathrm{e}(_{\mathcal{S}})$ 严重坍缩归零，其自然对数 $\mathrm{o}\,\mathrm{e}$ 趋近于 。
- **窗口控制解法：** 与 MMR 类似，利用滑动窗口 实时截断旧历史，使得优化式自适应转化为:

$$\mathrm{r}_i \cdot \mathrm{rer}_i + (1 \ ) \cdot \mathrm{o}\,\mathrm{e}(_i)$$



4. 强硬约束与重排业务规则

工业级典型强插规则

在线上推荐生态中，算法纯粹产出的多样性通常无法完全覆盖多元化的运营与产品设计诉求，重排系统还需强制执行硬性业务漏斗：

1. **类型连续性限制：** 控制相同体裁连续高频轰炸（例如：图文和视频笔记最多各自连续透出 $k=$ 篇；若超过此阈值，下一个位置必须强行轮换另一种体裁）。
2. **商业化曝光间距：** 控制商业化或运营推广笔记的透出频次（例如：运营推广笔记的精排分数会乘以一个大于 1 的 `boost` 系数以扶持其曝光，但重排强制规定每 $k=$ 篇笔记内最多只能掺杂 1 篇此类扶持内容）。
3. **关键位置截断限制（首屏保护）：** 离终端用户视觉冲击最强的首屏特定位置（Top-）必须实施更严苛的控制（例如：前 $=1$ 篇不允许出现带电商卡片的笔记，前 $=$ 篇首屏笔记中最多仅允许塞入 $k=1$ 篇电商转化卡片）。

规则与多样性算法（MMR/DPP）的联合求解

- **动态可行子集剔除：** 在每一轮贪心状态转移前，系统首先启动“业务规则过滤器”，将当前不满足硬性产品规则约束的候选物品强行剔除，过滤生成一个动态合规的子集 。
- **子集内寻优：** 裁剪后，将经典算法中的全集 无缝替换为可行子集 ，在此子集内部寻找最大化多样性指标的分数项，即可实现“业务硬约束”与“用户偏好多样性”的完美协同落地：

$$\mathrm{r}_i \cdot \mathrm{rer}_i + (1\ ) \cdot \mathrm{erScore}(i\,)$$

💡 Key Takeaways

推荐系统的重排阶段致力于解决精排 Pointwise 点分制下候选内容同质化堆叠的严重工程痛点，核心在于动态权衡单品价值与列表多样性。通过深入发掘基于 CLIP 图文多模态对比学习的内容级表征向量，系统能够获得不依赖双塔召回的更纯粹的物品间语义相似度。在具体执行层，业务规则过滤器可以动态裁剪出可行物品子集，并配合引入了时间滑动窗口的 MMR 或经由 Cholesky 增量分解加速的 DPP 几何行列式算法，最终在线上极低时延下完美实现了业务强硬约束、用户即时兴趣与信息茧房破局的协同演进。

VII. 物品冷启动

1. 冷启动评价指标

核心定义与优化目标

- **冷启动对象：** 主要针对平台内新发布、缺少用户交互数据的 UGC 内容（如小红书新笔记、B站新视频、今日头条新文章）。
- **优化三大目标：**
  1. **精准推荐：** 克服低交互难题，将新物品准确分发给合适用户，保障基础体验。
  2. **激励发布：** 流量向低曝光新物品倾斜，增强作者发布积极性，扩大内容池。
  3. **挖掘高潜：** 通过初期小流量进行试探性探索，发掘高质量内容并追加流量倾斜。

三大维度评价指标与公式

- **作者侧指标（反映创作积极性）：**
  - **发布渗透率 (Penetration Rate)：** 当日发布人数占日活人数的比例（当日发布 1 篇及以上即计入发布人数）。

$$\text{发布渗透率} = \frac{\text{当日发布人数}}{\text{日活人数}}$$

  - **人均发布量：** 当日总发布物品数与日活人数的比值。

$$\text{人均发布量} = \frac{\text{当日发布笔记数}}{\text{日活人数}}$$

- **用户侧指标（新内容消费表现）：**

    - **新物品消费指标：** 关注新内容的点击率和交互率。由于曝光具有极高的**基尼系数**（少数头部内容占据了大部分曝光），因此需要隔离并分别考察高曝光（如 >1000 次）与低曝光（如 <1000 次）的新内容。
    - **大盘消费指标：** 包含大盘总消费时长、日活（DAU）和月活（MAU）。大力扶持低曝光新内容通常会以牺牲短期大盘消费指标为代价。
- **内容侧指标（挖掘能力考核）：**
    - **高热笔记占比：** 指新内容在前 30 天内获得 1000 次以上点击的笔记比例。该指标越高，证明冷启系统挖掘优质内容的能力越强。

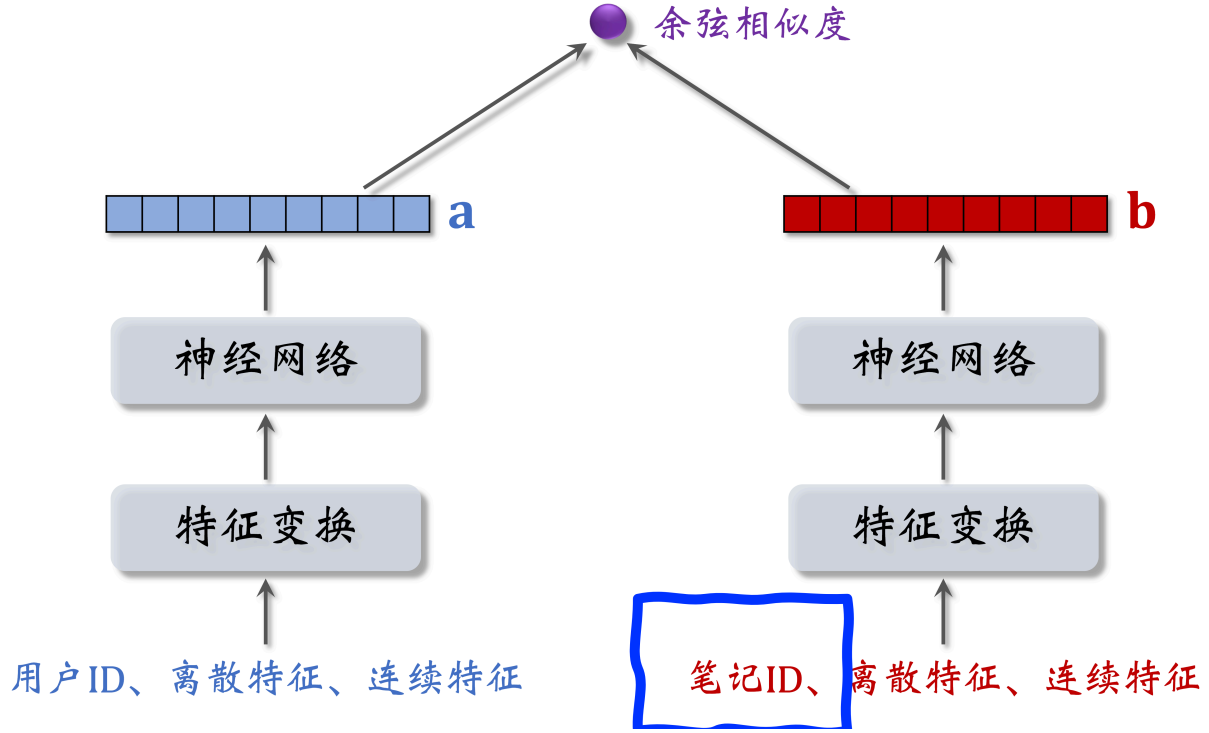
2. 冷启召回通道与网络简单改造

经典召回的冷启痛点

- **ItemCF 失效：** 新物品极度缺乏交互行为，无法建立起物品与物品之间的协同过滤关联。
- **双塔模型缺陷：** 新物品在冷启阶段尚未学好专属的 ID Embedding，直接线上检索会导致效果极差。

双塔模型改造方案

- **方案一：使用 Default Embedding**
    - 在物品塔做 ID 编码时，让全量新物品在初期强制共享同一个默认 ID（Default ID），并输出相同的共享向量表征。直到下一次离线模型训练更新时，新物品才正式拥有自己真正的 ID Embedding。
- **方案二：利用相似高曝内容均值（高曝平替）**
    - 寻找 Top-k 个在多模态内容上与该新物品最相似的高曝光老物品，将它们的 Embedding 向量取平均值，直接作为该新物品的初始 ID 向量表征。
- **多窗口召回池建设：**
    - 构建“1小时新笔记”、“6小时新笔记”、“24小时新笔记”、“30天笔记”等多个基于时间的隔离召回池，赋予新物品更多曝光机会。由于它们共享同一个双塔模型，因此并不增加额外的训练代价。



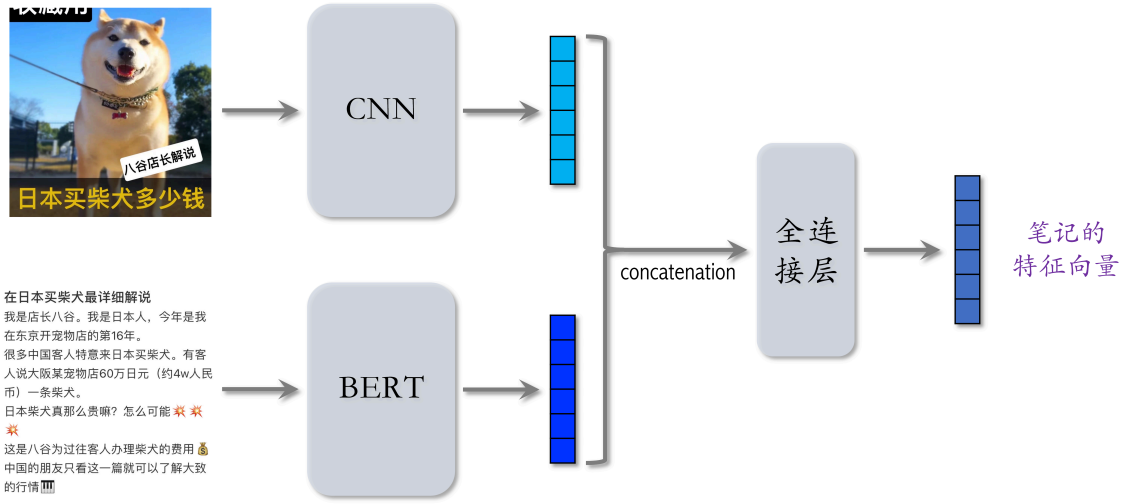
类目、关键词召回

- **类目与关键词反向倒排召回：**
    - 系统维护 `类目/关键词 -> 物品ID列表（按时间倒排）` 的反向索引。根据用户画像的兴趣标签，直接取回对应最新的 k 篇物品。
    - **局限性：** 时效窗口短（发布几小时后即被新内容冲刷无缘召回），且属于弱个性化，推荐精度不高。

3. 聚类召回

核心思想与内容表征网络

- **核心泛化逻辑：** 假设用户喜欢某篇历史笔记，则其大概率也会喜欢内容或语义相似的新笔记。
- **多模态特征提取：** 线下训练一个深度多模态模型：
  - **视觉表征：** 图像内容通过 CNN 提取特征。
  - **文本表征：** 文字标题与正文通过 BERT 提取语义。
  - 融合两者进行特征拼接（Concatenation），经过全连接层（FC）隐式投射，生成一个代表物品纯语义的稠密特征向量。



聚类倒排索引构建

1. 离线使用 **K-means 算法**（基于余弦相似度）将全量物品向量划分为 1000 个 Cluster，并记录每个聚类中心的向量方向。
2. 新物品发布后，通过多模态网络映射，寻找与其最相似的聚类中心，并划归至该 Cluster 下。
3. 实时维护高效索引：`Cluster -> 物品ID列表（按时间倒排）`。

线上召回与损失函数设计

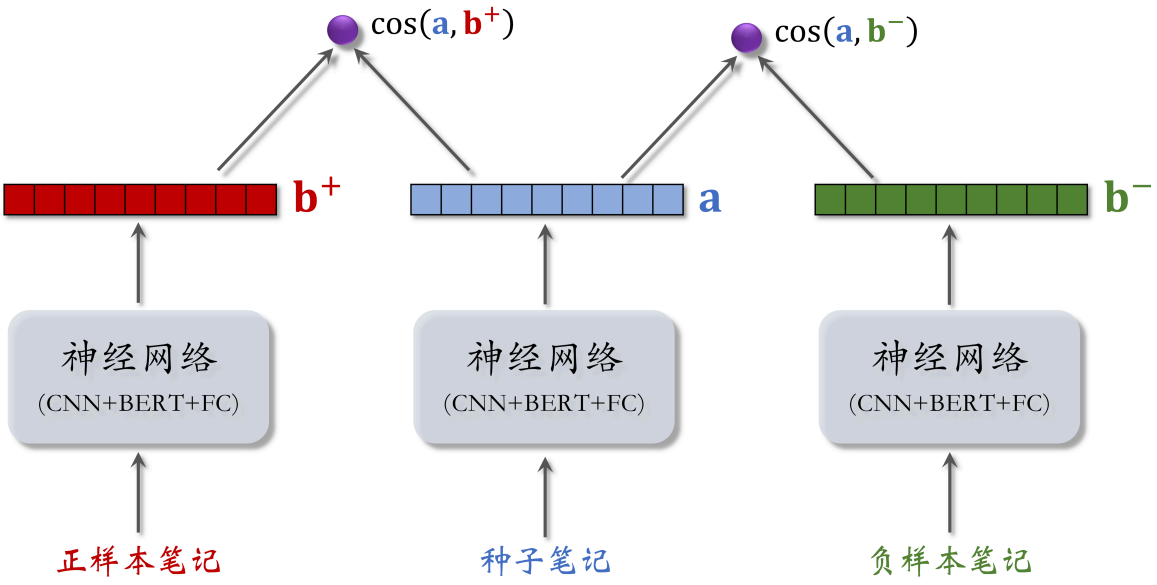
- **线上流程：** 输入用户 ID → 获取其 Last-n 次交互的种子物品 → 将种子物品通过网络映射并检索对应的 Cluster 中心 → 从这些 Cluster 列表中直接截取最新的 篇新内容。单次召回上限为 n 篇。
- **多模态网络损失函数（Triplet Loss）：**
  - **Triplet Logistic Loss:**

$$L(a^+) = \mathrm{o}\left(1 + \mathrm{e}(\mathrm{co}(a)\ \mathrm{co}(a^+))\right)$$

  - **Triplet Hinge Loss:**

$$L(a^+) = \left(0\,\mathrm{co}(a) + \ \mathrm{co}(a^+)\right)$$

  - *样本筛选机制：* 仅挑选交互充沛的高曝内容作为三元组，要求正样本具有相同的二级类目（如都是“菜谱教程”）并借助 ItemCF 相似度辅助圈定；负样本则从字数多、质量高的高曝池中随机抽取。
- **模型的训练：**
  - **基本想法:** 鼓励$s(a^+)$大于$s(a)$.



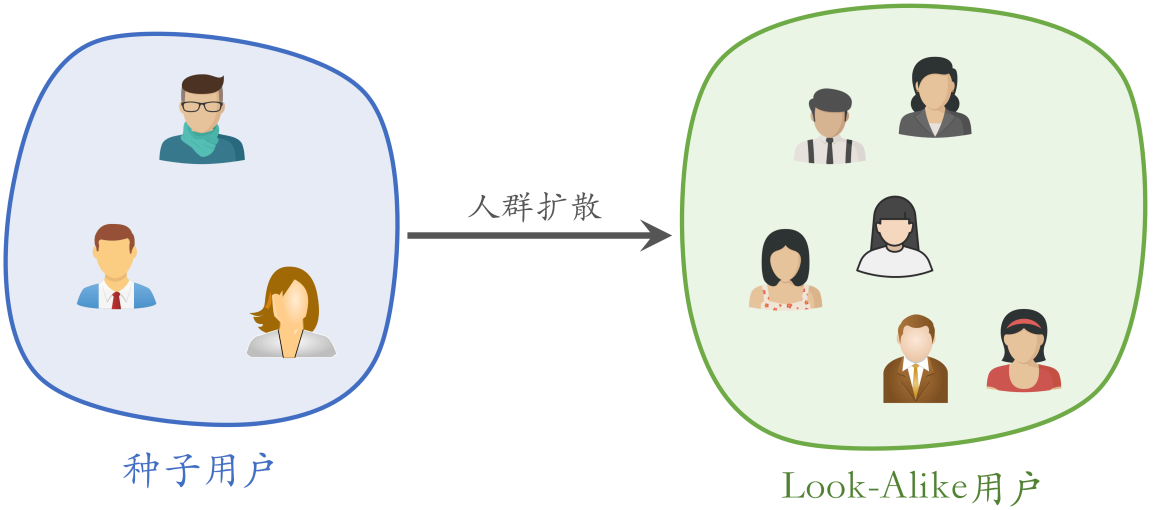
Remark:

注意到，在训练模型时，可以主动mask掉一部分笔记的数据作为种子笔记。利用ItemCF，得到理想的正样本和负样本数据。

4. Look-Alike 人群扩散

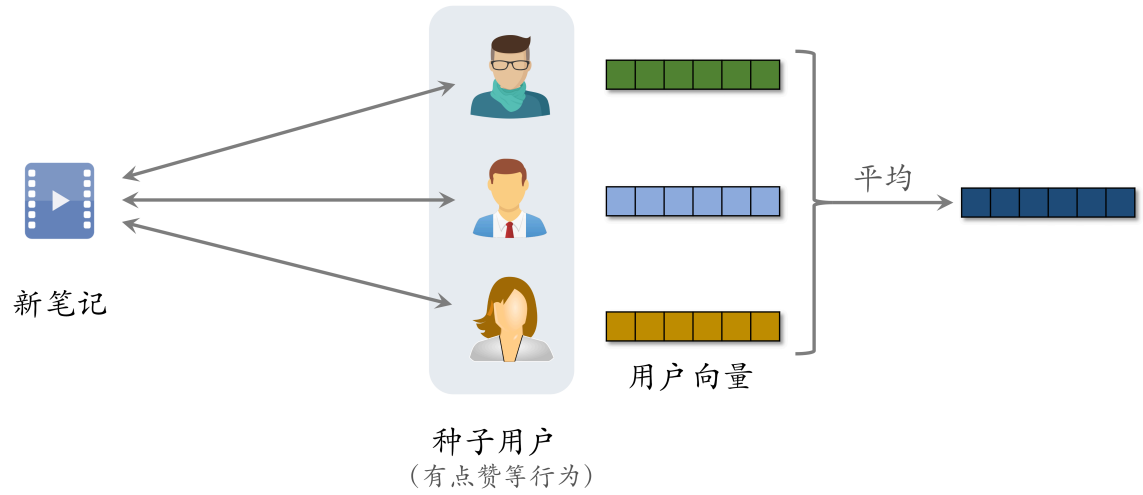
Look-Alike 扩散机制

- **核心概念：** 衍生自互联网广告的人群定向技术。将已经对新物品产生实质交互（点击、点赞、收藏、转发）的用户圈定为**种子用户**（Seed Users），通过算法在更广阔的用户池中寻找与种子用户特质相似的人群进行扩散分发。



- **近线特征向量更新：**
  - 新物品没有稳固的 ID Embedding，其代偿特征向量直接通过**动态计算已交互种子用户向量的平均值**来隐式表达：

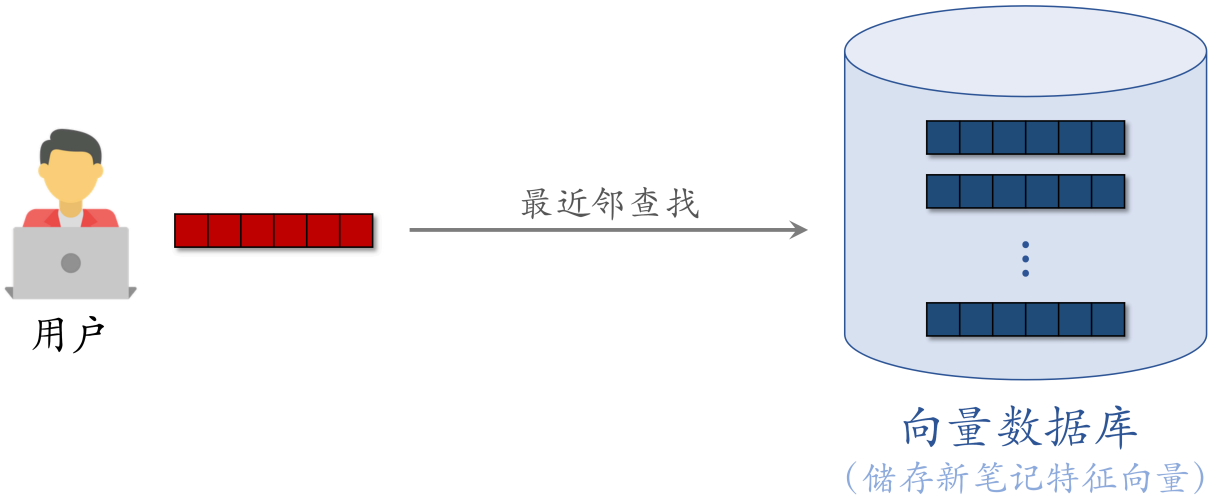
$$e = \frac{1}{ee} u_{ee} u$$



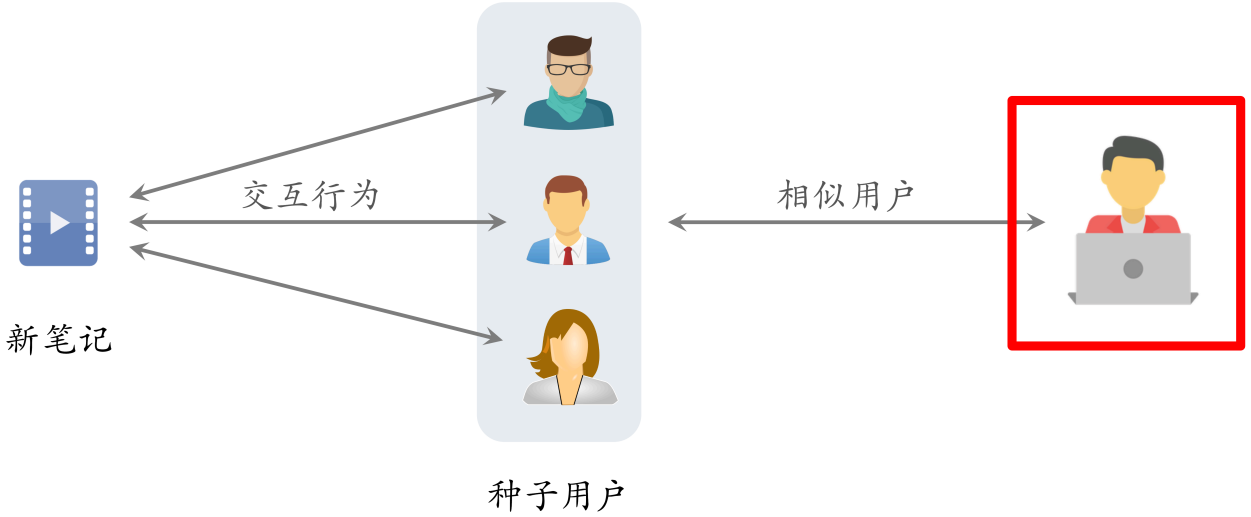
- 系统采用近线（Near-line）架构，每当有新用户对该内容产生交互行为，实时将其用户向量追加融合，实现新物品特征向量的自适应迭代演进。

线上检索检索闭环

- 动态更新的新物品特征向量会实时同步至**向量数据库**中。



- 线上分发时，以目标用户的 User Embedding 为查询基准，在向量数据库中进行反向最近邻查找（ANN Search），发掘与该用户兴趣最贴近的新内容，完成人群扩散闭环。



5. 流量调控

扶持新笔记的目的

1. 目的1：促进发布，增大内容池。
   - 新笔记获得的曝光越多，作者创作积极性越高。
   - 反映在发布渗透率、人均发布量。
2. 目的2：挖掘优质笔记。
   - 做探索，让每篇新笔记都能获得足够曝光。
   - 挖掘的能力反映在高热笔记占比。
3. 目的3：
   - 假设推荐系统值分发年龄$\leq$ 30天的笔记。
   - 假设采取自然分发策略，新笔记（年龄$\leq$ 24小时）的曝光占比约1/30，不利于平台优质内容挖掘。

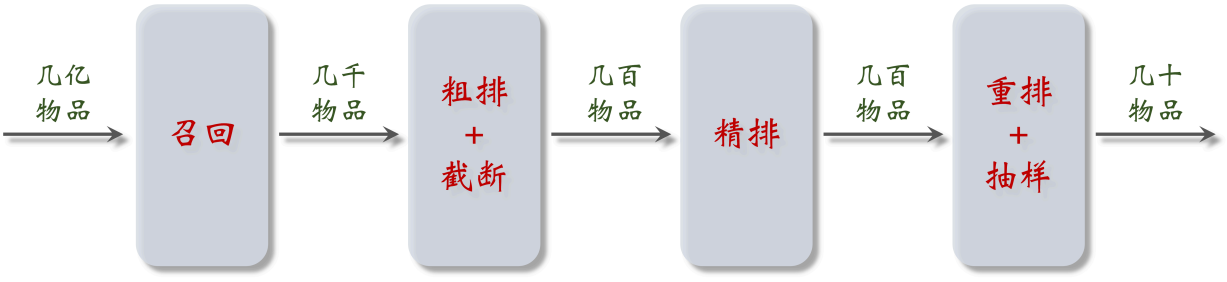
流量调控技术演进阶段

1. **结果硬插：** 直接在推荐结果页面的特定位置强行插入新物品。
2. **打分提权 (Boost)：** 在粗排和重排环节，直接将新内容的排序预估分乘以一个人工干预的提权系数：$\text{Score}_e = \text{Score}\ \text{oo}$。
   - **优缺点:** 极易实现、ROI 高；但曝光量对提权系数极其敏感，极难精准控量，易出现过度曝光或不充分曝光。
3. **动态提权保量：** 设定硬性曝光底线（如保证全量新物品在前 24 小时必须达成 100 次曝光）。

		当前曝光次数			
		0~24次	25~49次	50~74次	75~100次
发布时间	0~5小时	1.0	1.0	1.0	1.0
	6~11小时	1.1	1.0	1.0	1.0
	12~17小时	1.2	1.1	1.0	1.0
	18~24小时	1.3	1.2	1.1	1.0

注：不是小红书的真实数据

4. **差异化保量：** 摒弃大锅饭模式，根据内容质量与作者历史信用表现，分级别、精细化地设计保量目标。



> **Remark:**
>
> 人工给新笔记分数提权（Boost）主要干预粗排和重排环节。

动态保量公式与调节难点

- **动态提权系数调节函数：** 提权系数是由发布时间、目标时间、已有曝光及目标曝光共同决定的四元动态系统：

$$\text{提权系数} = f\left(\frac{\text{发布时间}}{\text{目标时间}}, \frac{\text{已有曝光}}{\text{目标曝光}}\right)$$

  - *实例：* 设定 24 小时需保量 100 次。对于已发布 18-24 小时但曝光数仍卡在 0-24 次的低曝光笔记，提权系数将自动拉满（如 1.3 ）以疯狂补流；而提前达标的笔记则系数回落至 1.0。
  - **保量的难点:** 很多笔记在24小时达不到100次曝光。保量成功率远低于100%。
    1. 召回、排序模型存在不足。
    2. 提权系数调得不好。
    3. 线上环节发生变化（i.e. 新增召回通道、升级排序模型、etc.）
- **保量的两难痛点（核心思考）：**
  - 为什么不能直接给新内容拉满超高提权（如 4 倍）直到满足 100 次曝光?
  - *根本原因：* 分数过度提权（Boost）会强行将新内容分发给不匹配的非目标受众，直接引发点击率、点赞率大跌。冷启结束后，这些充满负反馈的低劣表现会触发推荐系统底层大盘算法的长期打压，使其彻底丧失成长为全网热门笔记的潜力。

差异化保量体系

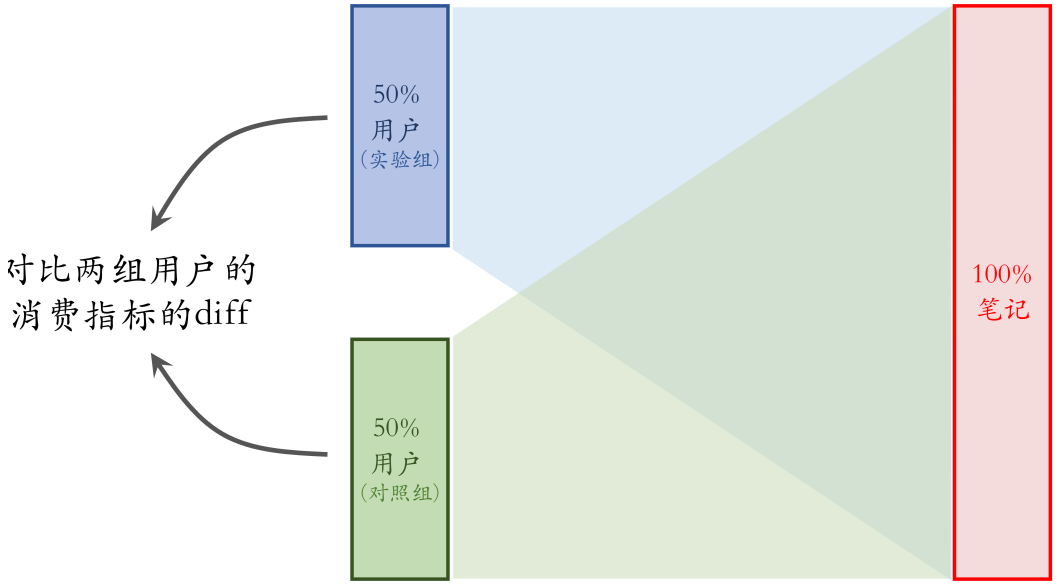
- **基础目标：** 24 小时保量 100 次曝光。
- **额外加成奖励：**
  - **内容质量加成：** 通过离线多模态内容模型评估内容质量，优质内容最高额外追加 +200 次曝光目标。
  - **作者质量加成：** 根据创作者历史笔记的平均表现及信用评分，最高追加 +200 次曝光目标。
  - **最终分级目标：** 彻底打破大锅饭，使不同单品的新内容差异化地分布在 100 00 次曝光的科学保量区间内。

> **Remark:**
>
> 可以通过多模态图文神经网络学习笔记内容给笔记内容质量预打分以实现查喜欢保量体系。

6. 冷启动阶段的特殊 AB 测试

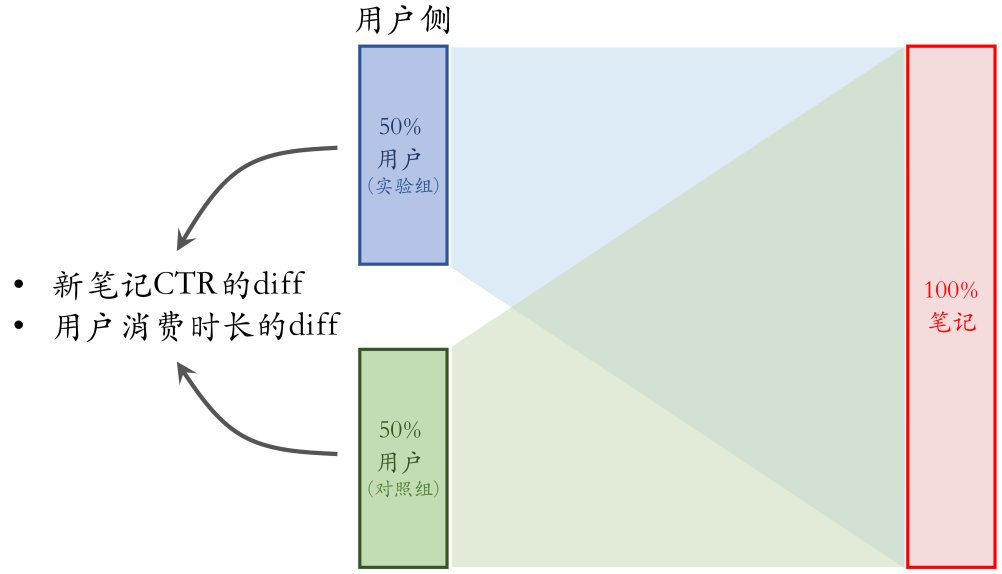
推荐系统标准的AB测试

推荐系统标准的AB测试



冷启动阶段的用户侧AB测试

- **实验设计方案：**



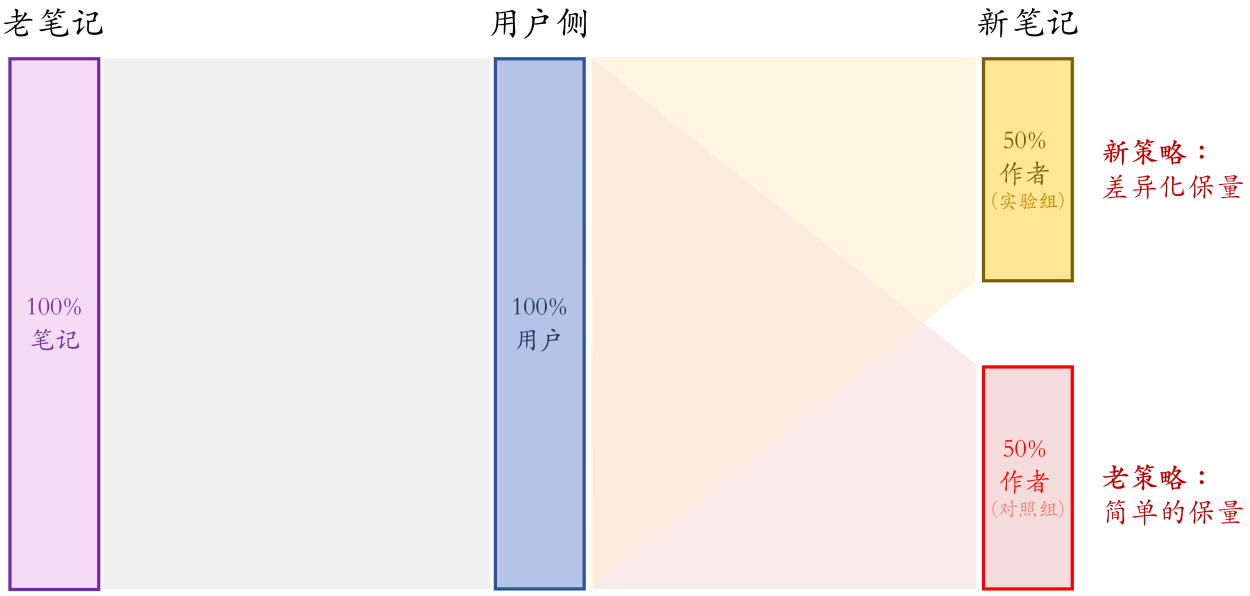
- **潜在缺点：**
  - 限定：保量100次曝光。
  - 假设：新笔记曝光越多，用户使用APP时长越低。
  - 新策略：把新笔记排序时的权重放大两倍。
  - 结果（从消费指标角度）：
    - AB测试的diff是负数（实验组不如对照组）。
    - 如果推全，diff会缩小（i.e. -2% → -1%）。

冷启动阶段的作者侧AB测试

由于冷启动策略直接作用于作者侧或内容侧，采用传统用户侧（User-side）均匀切分流量的测试方法会引发严重的数据失真。

方案一：队列隔离与作者侧切分

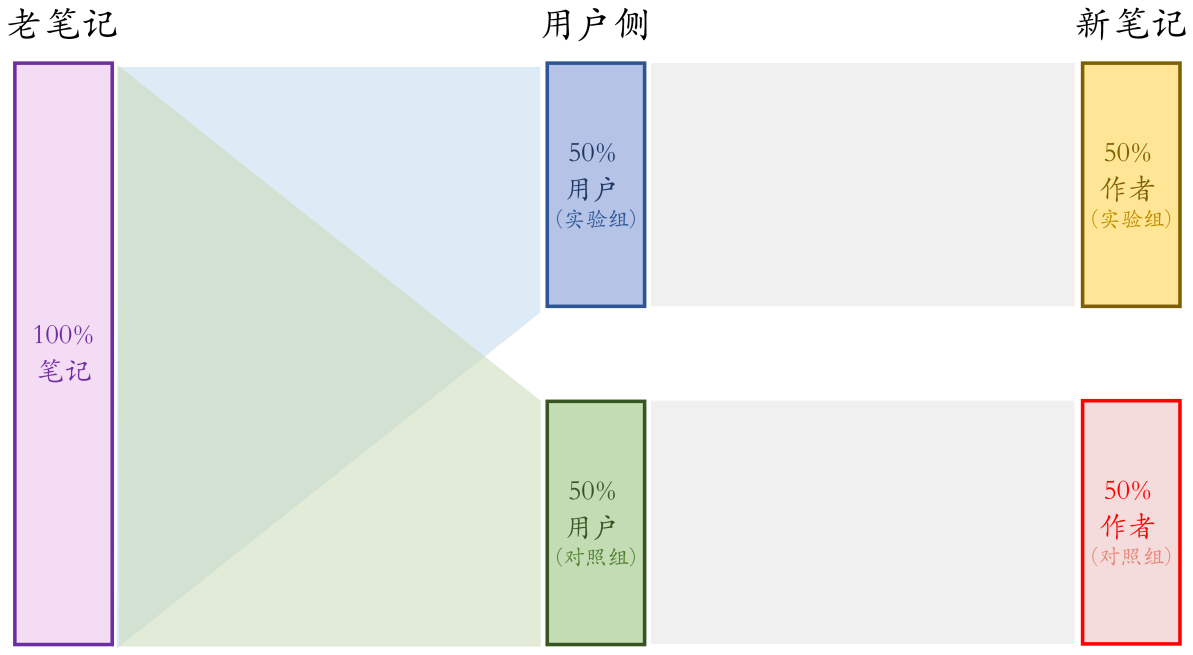
- **实验设计：**



- **设定：** 强行取消竞争，新老物品走各自的独立分发队列，锁定大盘重排流量分配（如新物品锁死 1/3 流量，老物品 2/3）。在作者侧将作者划分为 50% 对照组和 50% 实验组（带提权策略）。
  07_ColdStart_06, p.10
- **缺点（新物品内卷抢流量）：** 在 AB 测试期间，实验组能看到显著的正向 Diff。但由于 1/3 的流量比例是硬性死锁的，**一旦全量推全，该正向 Diff 会立刻缩减甚至彻底消失**。因为测试期间策略组只是抢占了对照组新物品的流量，并未能真正突破隔离墙抢到老内容的流量。

方案二：自由竞争与全局切分

- **实验设计：**



- **设定：** 拆除新老物品之间的隔离队列，允许新老内容利用 Boost 提权系数在全局打分中同场肉搏、自由竞争。
- **缺点（新老内容混战）：** 在 AB 测试期间，只有 50% 的新内容带着提权策略去和 100% 的老内容抢流量。而一旦全量推全，则是 100% 的新内容同时带着提权策略去向老内容发起冲击，这依然会导致测试阶段的指标反馈与最终推全的实际大盘结果产生极大的偏离。

💡 Key Takeaways

物品冷启动是 UGC 社区平台生态建设的核心一环，其核心目标是在保障用户精准体验与挖掘大盘高潜笔记的同时，通过流量倾斜最大化激励创作者的发布积极性。在召回阶段，系统必须依靠 Default Embedding 改造、图文多模态特征下的 K-means 聚类索引召回以及动态计算交互均值的 Look-Alike 人群扩散，来彻底攻克新内容缺乏 ID 向量与 ItemCF 行为数据塌缩的底层痛点。在流量调控及验证阶段，系统依靠融入内容和作者双重声誉模型的差异化保量算法进行精细化动态提权，并针对冷启策略在 AB 测试中容易引发的“新老内容互抢流量”及“推全 Diff 消失”等局限进行了多队列、自由竞争等深度的架构性方案攻坚。

VIII. 涨指标的方法

1. 推荐系统核心指标

- **核心指标 (Core Metrics)：**
  - **日活用户数 (DAU) 与留存 (Retention)：** 推荐系统最核心的终极指标。工业界最常用 **LT7** 和 **LT30**（衡量某用户在未来 7 天或 30 天内登录 APP 的实际天数）来评估留存长短。其中 $1 \le LT7 \le 7$，$1 \le LT30 \le 30$，且 LT 的增长通常直接对应用户体验的提升。
    - **用户使用时长、总阅读数（总点击数）、总曝光数：** 其重要性次于 DAU 和留存。时长增长时，LT 通常同步增长，但总阅读数和总曝光数可能反而会下降。
- **非核心指标 (Non-core Metrics)：** 单纯的点击率（CTR）、单次交互率等。
- **UGC 平台特有核心指标：** 创作者侧的发布量、发布人数以及发布渗透率。
- **核心优化五大方向：**
  1. 改进与添加召回模型；
  2. 升级粗排与精排架构；
  3. 提升全链路多样性；
  4. 针对新用户与低活用户进行特殊人群治理；
  5. 深度利用关注、转发、评论等高阶社会化交互行为。

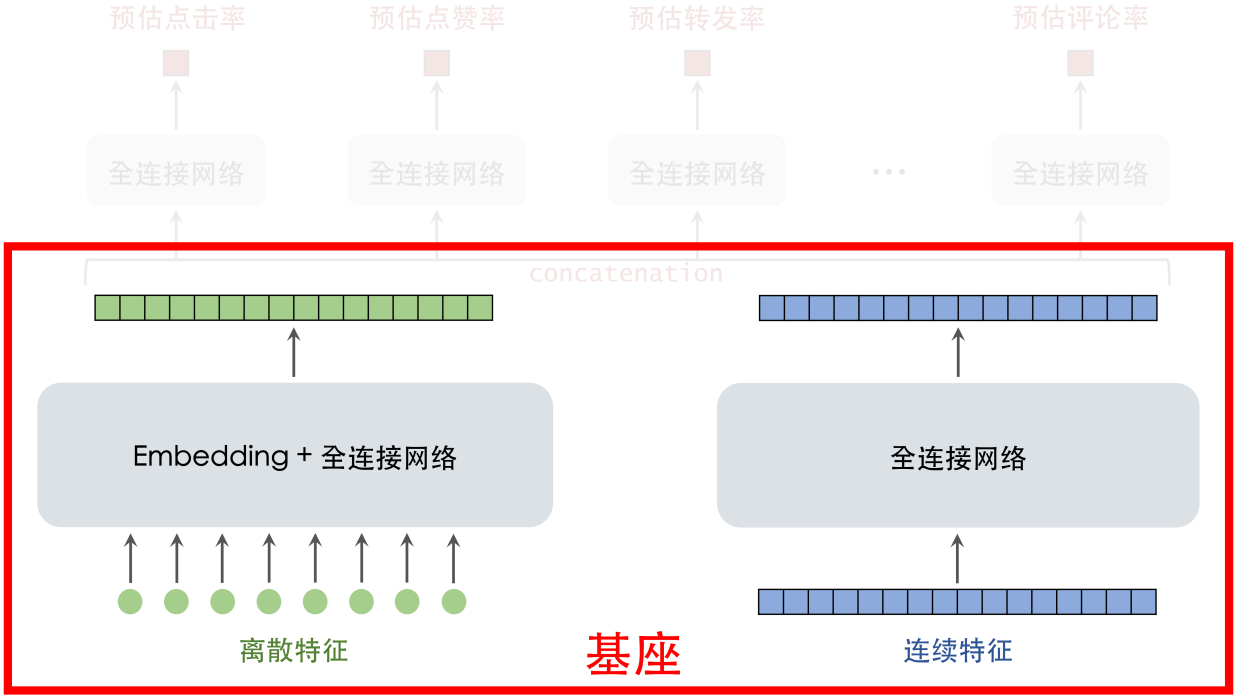
2. 召回通道优化

- **配额控制与多内容池分流：** 推荐系统通常并行几十条召回通道，总召回量配额固定。在总量不变时，通过合理调整各用户群体的配额并划分精细的内容池（如 30天/1天/6小时时效池、新用户优质池、分人群专属池）能有效拉升指标。
- **双塔模型（Two-Tower）三大迭代维度：**
  1. **样本工程优化：** 精细划分简单正样本（点击二元组）、简单负样本（随机未点击组合）与困难负样本（模型排序靠后但用户未点击的负反馈）。
  2. **网络结构改进：** 使用 DCN（深度交叉网络）替代经典的全连接网络（MLP）；在用户塔引入长短期行为序列（Last-n）；采用多向量模型打破单向量表达的上限瓶颈。
  3. **训练机制升级：** 引入二分类与 Batch 内负采样相结合的损失函数（需进行流偏纠偏）；采用自监督学习技术，强行让长尾/冷门物品的 Embedding 向量学得更稳健。
- **Item-to-Item (I2I) 召回升级：**
  - 最经典路径为 **U2I2I**（用户 $\rightarrow$ 历史产生过交互的 Item $\rightarrow$ 计算所得的相似 Item）。
  - **相似度矩阵计算：** 线上同时并行 ItemCF、Online ItemCF、Swing、Online Swing 等基于用户重合行为的协同过滤变体并各分配一定配额；或者基于双塔/图神经网络隐式产出的内容向量直接计算余弦相似度。
- **小众与泛化召回模型：**
  - 引入类似 I2I 的多级跳跃泛化：**U2U2I**（相似用户兴趣扩散）、**U2A2I**（喜爱作者新品召回）、**U2A2A2I**（相似作者物品扩散）。
  - 引入复杂网络：**PDN**（基于路径的深度网络）、**Deep Retrieval**（端到端检索结构）、**SINE**（稀疏兴趣网络）以及 **M2GRL**（多任务多视角图表征学习）。

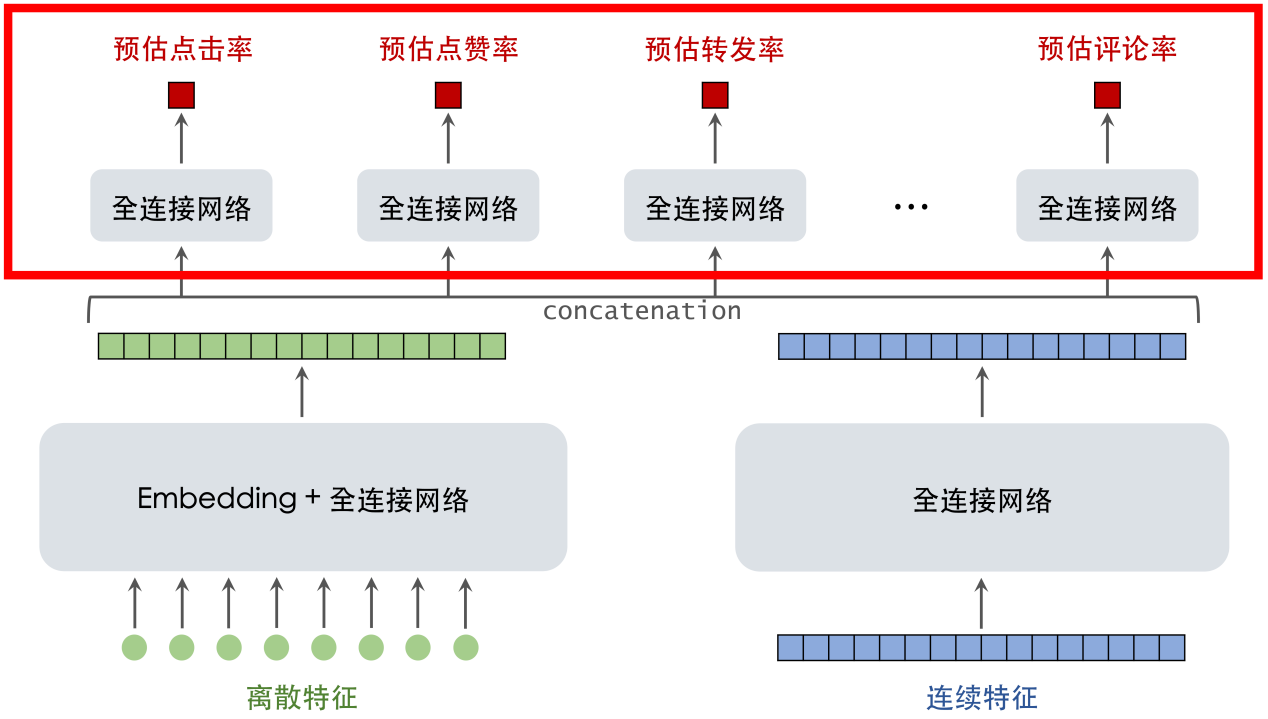
3. 排序模型迭代

精排模型升级

- **基座网络（Backbone）改造：** 输入离散与连续特征进行拼接（Concatenation）输出稠密向量，作为下游多目标的输入。改进策略包含：将基座网络加宽加深以提升预测精度；引入 **Bilinear**（双线性）或 **LHUC** 等精细化的自动特征交叉层；融入多模态内容特征与高维统计特征。



- **多目标预估（Multi-Task）改进：** 基础网络基于统一基座同时输出点击率、点赞率、转发率、评论率等多维度概率预测。进阶升级包括加入更丰富的交互新目标、升级为 MMoE/PLE 架构，并显式建模以消除位置偏置（Position Bias）。



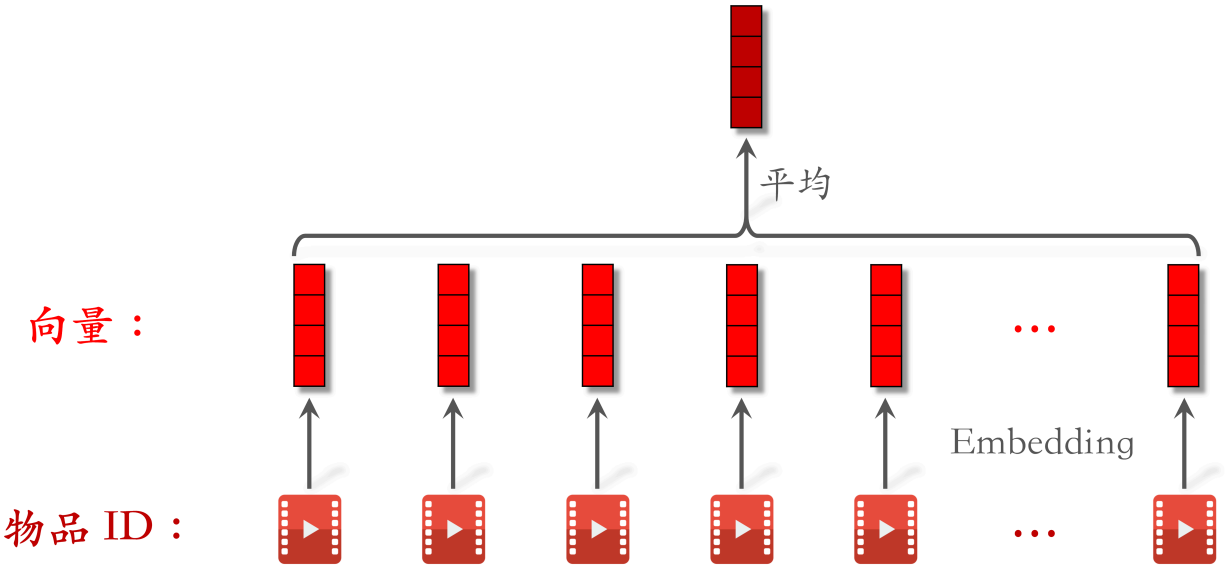
粗排模型与一致性建模

- **结构演进：** 粗排打分吞吐量比精排大 10 倍，首要要求高速度。工业界正逐步从传统的多向量双塔向效果更好但工程实现难度较大的 **COLD 三塔模型**演进。

- **粗精排一致性（知识蒸馏）：**
  - **Pointwise 蒸馏：** 粗排不直接去拟合真实交互标签 y，而是拟合精排输出的预估概率 p（例如当用户产生点击 $y=1$ 且精排预估 $p=0.6$ 时，粗排以平滑后的 $p_{new}=0.8$ 为目标进行逼近）。
    - **Pairwise / Listwise 蒸馏：** 引入排序学习（LTR）机制，强行让粗排去拟合精排对候选集合输出的**相对先后顺序**（例如若精排输出 $p_i > p_j$，则粗排若输出逆序 $q_i \le q_j$ 将触发 Pairwise Logistic Loss 惩罚）。

用户行为序列建模

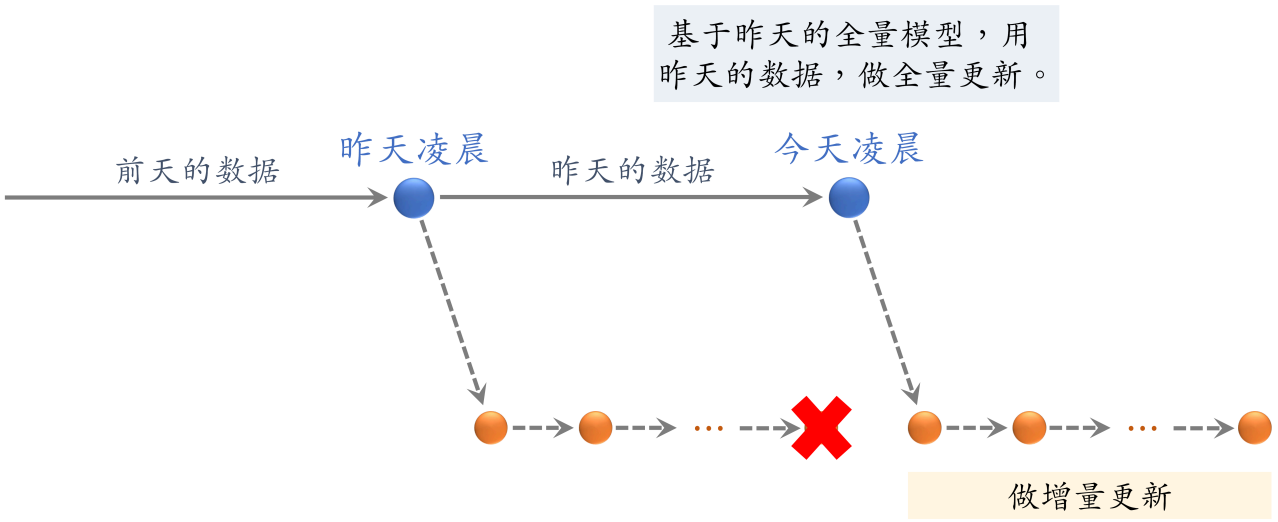
- **技术演进：** 经历了从最简单的 Mean Pooling（取平均作为特征）到引入注意力机制的 **DIN**，再到目前工业界主流的 **SIM**（搜索机制）架构的演进。

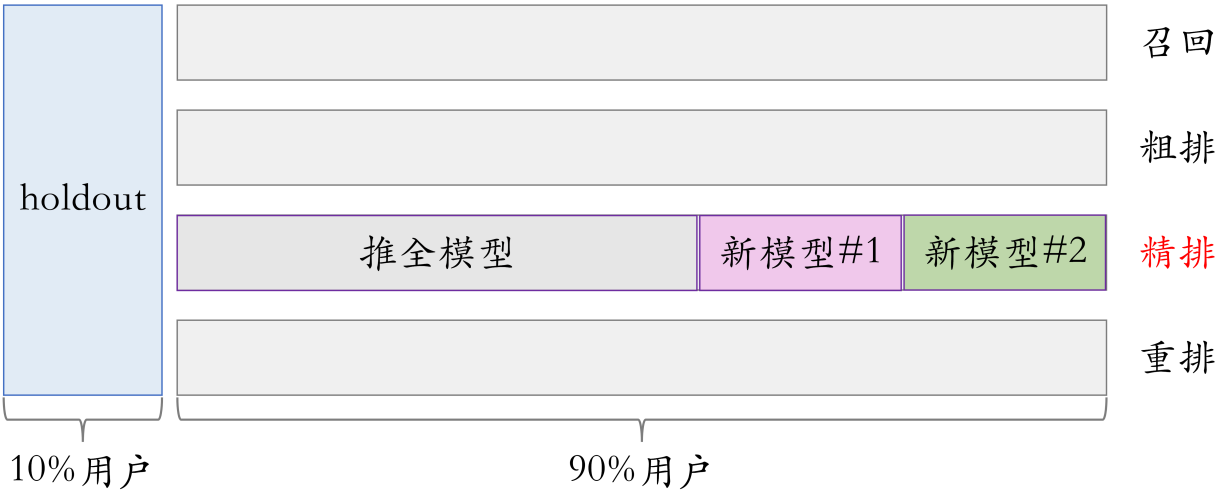


- **超长序列优化（SIM 迭代方向）：**
  - **序列长度扩展：** 延长序列长度以精确捕捉长期不退化的兴趣，但需要深度优化推理时间成本。
  - **多维度实时筛选：** 面对百万级超长行为，除采用传统的类目硬匹配（Hard Search）外，利用离线多模态网络将物品特征向量聚类为 1000 类。线上排序时，仅筛选保留与当前候选物品聚类序号一致的行为物品（将百万级降至数千个），并取多种筛选路径的并集送入 DIN 注意力层。

在线学习与老汤模型治理

- **在线学习 (Online Learning)：** 采用全量或流式增量实时更新对指标提升巨大，但会增加系统负担并制约常规模型的冷启动迭代效率。





- **老汤模型 (Legacy Model) 快速追平技巧：**
  - **结构快速对齐：** 评估新结构时随机初始化 MLP，但直接复用老模型已训练成熟的 Embedding 参数，用 n 天数据跑 1 epoch 快速验证结构优劣。
  - **线上早日追平：**
    1. 尽可能多地复用老模型的 Embedding 层，避免从头初始化（Embedding 承载了系统对特征的记忆，学得比全连接层慢）；
    2. 将线上老模型作为 Teacher 展开知识蒸馏，用其预测值作为新模型的训练目标，实现快速渡过追赶期。

4. 多样性机制与特殊人群

全链路多样性控制

- **精排多样性（滑动窗口融合）：** 最终排序得分结合兴趣分数 s_i 与多样性差异分 d_i（Score $= s_i + d_i$），常采用 **MMR** 或 **DPP** 算法。精排直接决定最终曝光鄰近格子的排布，**必须配合滑动窗口**动态计算。
- **精排强打散策略：** 建立刚性类目打散（选中物品 i 后，后续 5 个位置禁止出现相同二级类目）与多模态内容打散（全库图文聚类为 1000 类，选中后 10 个位置禁止同属一类）。
- **粗排多样性：** 粗排对 5000 个物品打分时，前 200 个物品按照纯兴趣分 s_i 直接截断送入精排；剩余 4800 个物品全面转为根据 $s_i + d_i$ 多样性融合分排序并截取前 300 个送入精排，以此兼顾全局多样性。
- **召回多样性（双塔加噪与抽样）：**
  - **加噪法：** 在线上计算出 User Embedding 后、执行 ANN 检索前，往用户向量中主动注入随机噪声。用户历史互动的类目越窄（兴趣越单一），注入的噪声越强，以此强行打散召回边界。
    - **抽样法：** 针对行为序列展开非均匀抽样，平衡高频大类与长尾小类的基础占比。

特殊人群治理（新用户与低活用户）

- **痛点：** 行为极度稀疏，常规个性化模型严重有偏且极易流失，业务上应**以保留存为主**，不盲目追求消费量或探索。
- **三大治理动作：**
  1. **特殊内容池弱个性化召回：** 圈定特定高危人群（如 18-25 岁一二线城市男性），根据该人群交互次数与交互率离线筛选出高质内容池，提供高质量、弱个性化的召回弥补。也可引入因果推断（Causal Inference），根据物品对留存率的净贡献值来构建和更新该特殊内容池。
  2. **特殊排序保护策略：** 针对该特殊人群实行少出广告或不出广告的策略，且**绝对不在新/低活用户上做新发布物品的曝光探索（Boost）**，避免由于推荐不准伤害新客体验；探索行为应完全锁死在活跃老用户群体上。

3. **差异化融分公式与坑位死锁：** 由于低活用户点击稀疏（无点击即无后续交互），在融分公式中应大幅度提高预估点击率（CTR）的权重系数。精排策略上直接锁定前几个核心坑位（或 50 个结果中的 3 个黄金坑位）无条件留给预估 CTR 最高的物品，确保用户第一眼看到最想点的内容。

- **差异化的排序模型：**

  - 方法 1：大模型 + 小模型。
    - 用全体用户行为训练大模型，大模型的预估 p 拟合用户行为 y。
    - 用特殊用户的行为训练小模型，小模型的预估 q 拟合大模型的残差 $y-p$。
    - 对主流用户只用大模型做预估 p。
    - 对特殊用户，结合大模型和小模型的预估 $p+q$。

  - 方法 2：融合多个 experts，类似 MMoE。
    - 只用一个模型，模型有多个 experts，各输出一个向量。
    - 对 experts 的输出做加权平均。
    - 根据用户特征计算权重。
    - 以新用户为例，模型将用户的新老、活跃度等特征作为输入，输出权重，用于对 experts 做加权平均。

  - 方法 3：大模型预估之后，用小模型做校准。
    - 用大模型预估点击率、交互率。
    - 将用户特征、大模型预估点击率和交互率作为小模型（例如 GBDT）的输入。
    - 在特殊用户人群的数据上训练小模型，小模型的输出拟合用户真实行为。

5. 社会化交互与作者激励

- **促转发与外部 KOL 建模：** * 目标是在完全不损害站内大盘点击和基本交互的前提下，最大化吸引站外引流。
  - **识别方法：** 追踪用户历史转发行为在外部实际带回来的流量大小，以此精准筛选出具备跨平台引流能力的“站外 KOL”。
  - **排序融合：** 在精排融合公式中添加额外的增益项 $k_u \cdot p_{ui}$（其中 k_u 为该用户是站外 KOL 的权重，p_{ui} 为预估转发率），定向给这些 KOL 增加高概率转发内容的曝光。
- **粉丝数对发布渗透率的激励价值：**
  - UGC 平台将作者发布率作为核心指标，而产生关注和评论能够极大刺激作者的创作积极性。

    - **边际效应：** 作者历史积攒的粉丝数越少，每增加一个新粉丝对其创作积极性的边际激励效果就越大。
    - **低粉新作者扶持：** 定义一个关于作者粉丝数 f_a 的单调递减权重函数 $w(f_a)$。在排序融分公式中强行加入一项 $w(f_a) \cdot p_{ui}$（p_{ui} 为预估关注率），利用全局推荐流量帮助低粉和新创作者精准涨粉，全面激发其发布留存。
- **隐式关注关系挖掘：**
    - **显式关注：** 用户 u 显式点击了关注作者 a，通过 U2A2I 召回，其点击率与交互率指标在所有召回通道中表现最高。
    - **隐式关注：** 挖掘用户 u 历史上频繁消费、喜爱看某作者 a 发布的物品但未点击关注按钮的隐式依赖，同样可以利用此关系在 U2A2I 通道中进行强力召回。

💡 Key Takeaways

工业界推荐系统指标的综合增长依赖于全链路的精细化协同演进，核心围绕日活（DAU）和用户留存（LT）两大终极目标，通过在召回侧拓展双塔及 I2I 变体、在排序侧进行多目标基座加深与粗精排一致性蒸馏，实现技术底座的全面升级。系统通过在精排和粗排中引入 MMR/DPP 滑动窗口及聚类打散机制，并在双塔检索中对窄兴趣用户动态注入随机噪声，全方位保障了分发内容的多样性与探索价值。针对生态治理，系统一方面构建弱个性化高质量内容池并死锁高 CTR 坑位来保护新与低活用户留存，另一方面通过引入基于粉丝数边际效应的递减权重函数，在融分公式中向低粉及新作者进行流量倾斜，构建了健康的 UGC 社区双向互动生态。