推荐系统机器学习学习指南

副标题：从机器学习基础到推荐系统实践

版本：2026-10-30 第 2 版（初学者修订）

适用对象：会一点 Python，学过或正在补机器学习，希望亲手做出第一个推荐器的自学者。

学习主线：机器学习基础 → 数据处理与评估 → 协同过滤与矩阵分解 → CTR 预估 → 向量召回 → 深度排序 → 序列推荐 → 多目标与在线优化 → 图推荐与生成式推荐（选学）。

建议方法：先看场景，再算一个例子，最后尝试习题。每学一个模型，能说清它“吃进什么、吐出什么、靠什么学会、放在流程哪里”，比先记住名字更有用。

配套阅读：搜广推开源项目实战清单。这份指南讲“为什么”，清单帮你选“用什么动手”。参考资料与论文入口见附录 D，本次核查范围见附录 E。

目录

先别急着背模型：从给朋友挑电影开始

一、学习路线与基本约定

二、数学符号与公式阅读方法

第 1 章　推荐系统的任务与整体架构

第 2 章　机器学习基础：任务、损失、优化与泛化

第 3 章　数据、样本构造与特征工程

第 4 章　离线评估、数据划分与在线指标

第 5 章　基线、内容推荐与协同过滤

第 6 章　矩阵分解与隐式反馈推荐

第 7 章　CTR 预估：逻辑回归、FM 与树模型

第 8 章　Embedding、双塔召回与向量检索

第 9 章　深度排序模型：Wide Deep、DeepFM 与注意力

第 10 章　序列推荐与用户兴趣演化

第 11 章　学习排序：Pointwise、Pairwise 与 Listwise

第 12 章　多目标、多任务学习与概率校准

第 13 章　冷启动、探索利用与强化学习入门

第 14 章　重排、多样性、在线实验与反馈偏差

第 15 章　综合实践：从 MovieLens 到完整推荐 pipeline

第 16 章　进阶专题：图推荐、混合推荐、多场景与生成式推荐

附录 A　核心知识自测清单

附录 B　按完成情况推进的学习计划

附录 C　一句话知识索引

附录 D　参考资料、章节映射与原始论文

附录 E　本次修订与查证范围

附录 F　核心术语表

先别急着背模型：从给朋友挑电影开始

假设你要从一万部电影里，给朋友小林挑出今晚想看的 10 部。你大概会分三步：

1. **先圈一批可能喜欢的。** 小林最近看了几部科幻片，你先找科幻、相似观众喜欢的电影，再加一些热门新片。这是**召回**，像海选，先把一万部缩到几百部。
2. **再认真比较。** 同样是科幻，小林可能更喜欢太空探险而不是恐怖题材。你逐部判断，排出先后。这是**排序**，像面试，花更多精力看少量候选。
3. **最后检查整张片单。** 前 10 部如果全是同一个系列，可能太单调；你会去重、换几种口味。这是**重排**。

接下来，小林点了其中一部。但“点开”不一定表示“喜欢”：也许封面吸引人，看到一半却退出了。推荐系统既要学会挑内容，也要小心解释这些反馈。

先记住这条线就够了：**过去发生了什么 → 准备候选 → 给候选打分 → 组成列表 → 看反馈**。后面的公式和模型，都是在解决这条线上的某个问题。

第一次怎么读

不用先读完 16 章再写代码。建议分三趟走：

这一趟做什么	阅读范围	到这里就可以动手或暂停
第一趟：做出一个能推荐电影的小程序	第 1 章 → 第 2 章基本概念 → 第 3～5 章 → 第 15 章入门版	能跑热门推荐、ItemCF，算出 Recall@10，解释一次推荐
第二趟：让推荐器学会调参数	补第 2 章公式，再读第 6～9 章	在相同数据与规则下比较一个可训练模型和基线
第三趟：按兴趣选方向	第 10～14 章；第 16 章选学	在序列、广告、重排、图推荐中只选一个继续深入

每节可以分三层读：**先看场景和例子 → 再看公式在说什么 → 最后查参数表**。第一遍不懂推导，可以先标记，不必卡住整条路线。

开始前做个小检查：你能用 Python 的字典和集合存下“谁看过哪些电影”，能读懂 `[1, 2]` 与 `[3, 4]` 的点积是 $1 \times 3 + 2 \times 4 = 11$，就可以开始。Pandas 用于读表，神经网络留到第二趟补。普通笔记本和 CPU 足够完成第一个小项目；有 GPU 也先把小数据和评估做对。

> **今天就能完成的一步**：看完第 1 章，直接试第 5.3 节的小程序。你会先得到一份真实的推荐结果，再回来理解它为什么这么排。

一、学习路线与基本约定

1. 完整目录地图

这张表用于查找主题，不要求按顺序学完再做项目；第一次阅读按前面的“三趟路线”走，第 15 章入门版可以提前做。

阶段	内容	目标
第一阶段	第 1～4 章	理解推荐任务、机器学习基础、特征与评估
第二阶段	第 5～7 章	掌握协同过滤、矩阵分解和经典排序模型
第三阶段	第 8～11 章	掌握向量召回、深度排序、序列建模和学习排序
第四阶段	第 12～14 章	理解多目标、冷启动、探索利用、重排和在线实验
实践与选学	第 15～16 章	提前做第 15 章入门项目，后续按需读第 16 章

2. 常用符号

- 用户：u；物品：i；上下文：c。
- 用户—物品交互：(u, i)；真实标签：y；预测值：$\hat{y}$。
- 用户向量：p_u；物品向量：q_i。
- 模型打分：$s_\theta(u, i, c)$，其中 θ 表示模型参数。
- 推荐列表长度：K，例如 Recall@10 表示在前 10 个结果上计算召回率。注意：“召回阶段”是一个系统环节，“召回率”是一个指标，它们不是同一件事。

3. 推荐系统中的两类反馈

- **显式反馈**：评分、主动点赞等，能较直接地表达偏好，但通常稀疏，且评分标准因人而异。
- **隐式反馈**：点击、播放、购买、停留时长等，数据量大，但“未点击”不一定等于“不喜欢”，还可能因为物品未曝光或位置靠后。

4. 配套资料如何使用

四类资料承担不同职责，不建议混用：

资料	主要作用	使用方式
周志华《机器学习》	补齐监督学习、评估、集成、聚类、降维和强化学习理论	先读第 2、3 章；学深度模型时补第 5 章，其余按需查阅
南瓜书（PumpkinBook）	补足西瓜书中省略的公式推导	遇到理解困难的公式时按章查阅，不替代原书
fun-rec（小麦书）	对照工业级召回、排序、重排与生成式推荐体系	在完成本指南第 5～10 章后按专题阅读，并运行配套代码

资料	主要作用	使用方式
Aggarwal 教材	系统查阅经典推荐算法、评估、上下文与高级专题	作为百科全书式参考，按问题查阅而非顺序通读

使用原则：

1. **先建立问题框架，再读模型细节。** 每读一个模型，先回答它位于召回、排序还是重排，优化什么目标。
2. **教材与代码互相校验。** fun-rec 更新较快，公式、代码接口与章节可能随时间变化；遇到差异时以原始论文和可复现实验为准。
3. **不同资料的指标不可直接比较。** 数据、切分、候选集合和已见过滤规则不同，Recall@K 等数值没有可比性。
4. **论文用于核对定义。** 尤其在学习 BPR、NCF、Wide & Deep、DeepFM、DIN、SASRec 和 LightGCN 时，应回看原始论文的目标函数与实验设置。

5. 教材结构与学习闭环

为便于自学，正文采用统一结构：

- **章前**：学习目标、先修知识、本章导读，帮助你判断“要学什么”和“需要先补什么”。
- **正文**：概念讲解、关键原理、公式参数表、数值例子和常见误区。
- **章后**：本章小结、关键术语、习题及参考答案。
- **附录**：符号表、自测清单、学习计划、章节映射、原始论文、数据集和术语表。

建议每一章按以下闭环学习：

1. **预读 5～10 分钟**：只看学习目标、导读和本章小结，建立问题意识。
2. **精读正文**：先找公式的输入和输出，遇到陌生符号就查参数表，再用一句话解释它在算什么。
3. **手算一个例子**：至少复现文中的一个数值计算，不要只看结论。
4. **闭卷做题**：先独立写答案，再展开参考答案，重点分析答错的原因。
5. **用自己的话复述**：试着向同学解释一次推荐为什么出现。讲到卡住的地方，就知道下一次该补哪里。

二、数学符号与公式阅读方法

本指南面向已经接触过机器学习、但需要重新建立直觉的读者。为避免“公式看得懂每个字母，却不知道整体在做什么”，多数核心公式后都附有参数说明和直觉解释。阅读时可按以下四步进行：

1. **先看输出。** 这个公式算出来的是分数、概率、损失还是向量？
2. **再看输入。** 每个符号是用户、物品、特征、参数，还是训练超参数？
3. **确认维度。** 哪些是标量，哪些是向量，向量之间能否相乘？
4. **解释业务含义。** 数值变大表示更相关、更准确，还是惩罚更强？

1. 排版约定

- 斜体小写字母通常表示标量或编号，例如 u、i、t。
- 下标表示“谁的”或“第几个”，不决定它是否为向量。例如 p_u、q_i、h_t 是向量，但 b_u 是一个偏置数，r_{ui} 是一个评分。先看参数表中的类型。
- p_u^T 表示向量 p_u 的转置；$p_u^T q_i$ 是两个向量的内积，结果是一个标量。
- $\hat{y}$ 读作“y hat”，表示模型预测值；y 表示真实标签。
- 上标 (k) 表示第 k 层，不是 k 次方；例如 $e_u^{(k)}$ 是第 k 层的用户向量。
- 本指南中的 $\log$ 如无特别说明均指自然对数，底数为 $e \approx 2.718$。
- 同一个字母在不同章节可能承担不同角色，例如 K 既可表示前 K 个推荐，也可表示图网络层数。应以该公式后的参数表为准。

2. 全局符号表

符号	含义	类型或取值范围
u	用户编号	离散类别，例如用户 ID
i	物品编号	离散类别，例如商品 ID
c	上下文，例如时间、设备、页面	可以是类别或数值特征
n	样本编号	$1, \dots, N$
N	样本数量	正整数
j, k	通用索引，可表示特征、位置、物品或层	具体含义由局部公式说明
t	序列中的时间步	$1, \dots, T$
T	序列长度	正整数
y	真实标签	二分类中通常为 0 或 1
$\hat{y}$	模型预测值	回归任务可为实数，分类任务通常为 $[0, 1]$
r_{ui}	用户 u 对物品 i 的真实评分	由数据决定，例如 1～5
$\hat{r}_{ui}$	模型预测的评分或偏好分数	实数
s_{ui}	模型给用户 u 与物品 i 的匹配分数	实数，不一定是概率
θ, ϕ	模型可学习参数	向量、矩阵或参数集合
d	隐向量维度	由开发者设置的正整数
p_u	用户 u 的隐向量	d 维列向量
q_i	物品 i 的隐向量	d 维列向量
e_u, e_i	用户塔、物品塔输出的表示向量	通常为同一维度的列向量
x	特征向量	维度由特征设计决定
w	线性模型的权重向量；w_j 为第 j 个权重	向量；单个 w_j 为实数
b	偏置项	实数
λ	正则化或权衡系数	通常为非负实数

符号	含义	类型或取值范围
$\sigma(\cdot)$	Sigmoid 函数，把实数映射到 $(0,1)$	输出为概率形式
L	本指南常见损失函数的值；第 14 章局部用作列表名	这些损失通常非负，训练时希望减小
K	推荐列表截断位置，或图网络层数	正整数，按上下文判断
rel_j	第 j 个结果的相关性等级	二值或多级离散值
$\mathcal{N}(u)$	与用户 u 相连的邻居集合	集合
$\lvert\mathcal{N}(u)\rvert$	集合中邻居的数量	非负整数
$\lVert\cdot\rVert$	向量的 L2 范数，即各分量平方和再开方	非负实数
$\exp(\cdot)$	指数函数 $e^{(\cdot)}$	正实数
$\succ$	偏好或排序关系，$i \succ j$ 表示 i 应排在 j 前	关系符号
$\in$	属于某个集合	集合关系
$\sum$	求和	结果类型由被加项决定
$\top$	向量或矩阵转置	不改变数值，只改变排列

3. 常见公式的输出类型

- **打分公式**：输出实数，只保证大小关系，例如矩阵分解的 $\hat{r}_{ui}$。
- **概率公式**：输出 $[0,1]$ 之间的数，例如 Sigmoid 或 Softmax 的结果。
- **损失公式**：输出非负惩罚，训练目标通常是最小化它。
- **表示公式**：输出向量，用于后续计算相似度或输入下一层网络。

一个常见错误是把“打分”直接当“概率”。例如矩阵分解输出 3.0，并不表示 300% 的点击概率；只有经过明确的概率变换和校准，分数才可能具有概率含义。

第 1 章　推荐系统的任务与整体架构

> **本章学习目标**
>
> 1. 能用自己的话说明推荐系统解决什么问题，以及它与搜索、普通分类任务的差异。
> 2. 能画出“召回—排序—重排”的基本链路，并说明每个阶段输入什么、输出什么。
> 3. 能识别曝光偏差、反馈回路和离线指标与线上效果的差异。
>
> **先修知识**：不需要推荐系统基础；只需理解“概率”“特征”和“模型”的基本含义。
>
> **本章导读**：先建立全局地图，不急于记模型名称。后面每一章都会回到这张地图，说明自己位于哪个环节。

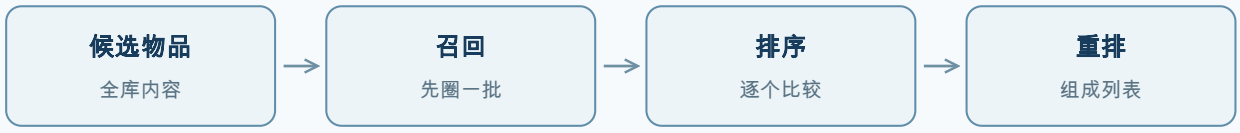
1.1 一万部电影，为什么只展示这十部

推荐系统的目标是：根据用户、物品和上下文信息，从大量候选物品中选出当前最值得展示的一组物品，并确定展示顺序。

典型工业链路包括：

1. **召回（Recall/Matching）**：从百万甚至更多物品中快速选出数百至数千个候选，重点是大规模检索能力与覆盖率。
2. **排序（Ranking）**：对候选进行精细打分，预测点击、转化、评分或观看时长等。
3. **重排（Re-ranking）**：在个体相关性之外处理去重、多样性、业务约束和列表整体效果。
4. **在线服务与反馈收集**：展示结果并记录曝光、点击等行为，供后续训练和评估使用。

1.2 海选、面试、检查片单，各管一件事



推荐列表的多阶段生成流程

推荐系统不是单一模型，而是多个目标不同的模块组成的系统：

- 召回强调“不要漏掉用户可能喜欢的物品”。
- 排序强调“把更可能产生目标行为的物品排在前面”。
- 重排强调“整个列表是否合理”，而非每个物品单独得分是否最高。
- 数据反馈会反过来影响未来训练数据，因此推荐系统存在反馈回路。

一个简化打分函数可以写为：

$$s_\theta(u, i, c) = f_\theta(x_u, x_i, x_c, x_{u,i})$$

公式参数说明

符号	含义	维度或取值
$s_\theta(u, i, c)$	模型在用户 u、物品 i、上下文 c 下给出的匹配分数	实数；按二分类目标训练后可经 Sigmoid 输出点击概率估计
u, i, c	当前请求中的用户、候选物品和上下文	离散编号或特征集合
x_u	用户特征向量，例如年龄、活跃度、历史偏好统计	维度 d_u，由特征工程决定
x_i	物品特征向量，例如类别、价格、热度、内容向量	维度 d_i，通常与 d_u 不同
x_c	上下文特征向量，例如小时、设备、页面位置	维度 d_c

符号	含义	维度或取值
$x_{u,i}$	用户与物品的交叉特征，例如“用户对该类目的历史点击率”	维度 d_{ui}
f_θ	把上述特征映射为分数的模型函数	可以是点积、树模型、MLP 或其他结构
θ	f 中所有可学习参数	权重、Embedding、树参数等

公式在说什么

这个公式不是要求把所有特征简单拼接，而是说明推荐分数由四类信息共同决定：用户是谁、物品是什么、当前场景是什么，以及用户与物品之间已经存在什么关系。f_θ 可以很简单，例如对特征做线性加权；也可以很复杂，例如先查 Embedding，再经过多层神经网络。

需要注意

s_θ 本身通常只是排序分数，不天然是概率。独立预测每次曝光是否点击时，常用 Sigmoid 加二分类交叉熵。只接一个 Sigmoid 并不会自动得到可信的概率，还取决于标签、训练数据与校准。Softmax 常用于“在给定候选中选下一物品”，输出依赖候选集合，不能直接当作各物品独立的曝光点击率；评分回归则通常保留实数输出。离线排序时，只要同一模型内分数可比即可；跨模型融合时，则必须检查分数尺度与校准状态。

1.3 换成短视频，流程还是一样

短视频推荐中：

- 召回：从内容库找出关注作者视频、相似视频、热门视频和向量近邻视频。
- 排序：预测每个候选的点击率、完播率和互动率。
- 重排：降低同一作者连续出现的次数，并插入一定数量的新内容。

1.4 常见误区与注意事项

- **误区：点击率越高，系统就越好。** 高点击率可能来自标题诱导，并不必然带来满意度或长期价值。
- **误区：离线指标提升一定会带来线上提升。** 离线数据受到旧策略影响，无法完全模拟新策略。
- **注意曝光机制。** 没有点击的物品可能根本没有被用户看到。
- **注意系统约束。** 延迟、候选数量、数据更新频率会直接影响模型可选方案。

本章小结

- 推荐系统是在曝光、库存、延迟等约束下，完成“候选匹配—精细排序—列表优化”的多阶段系统，而不是单一模型。
- 召回追求覆盖与效率，排序追求个体相关性，重排追求列表整体效用，三者目标不同。
- 隐式反馈充满噪声；“未点击”可能来自未曝光或位置靠后，不能直接等同于不喜欢。
- 线上反馈会改变未来数据分布，形成反馈回路。学习时先把离线评估做可靠；要判断真实线上收益，还需要设计合理的在线实验。

关键术语：召回、排序、重排、曝光、隐式反馈、反馈回路。

1.5 习题

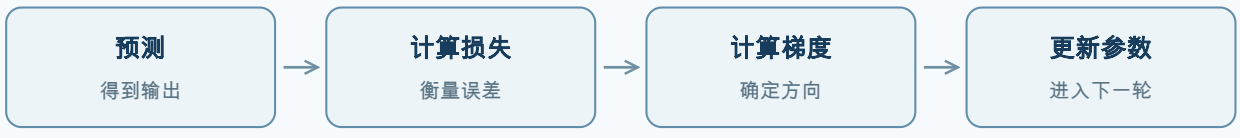
1. 为什么通常不在排序阶段直接对全库物品逐一使用复杂模型？
2. “用户没有点击某物品，所以用户不喜欢它”这一推断有什么问题？
3. 召回和重排分别更关注什么目标？

> **▼ 参考答案**
>
> 1. 全库规模通常很大，复杂模型逐个打分会产生无法接受的计算量和延迟；召回先用较低成本缩小候选范围。
> 2. 未点击可能由未曝光、展示位置靠后、标题不吸引或当时无需求造成，隐式负反馈含有噪声。
> 3. 召回关注候选覆盖和高效检索；重排关注列表整体质量，如去重、多样性和约束满足。

第 2 章　机器学习基础：任务、损失、优化与泛化

> **本章学习目标**
>
> 1. 能解释逻辑回归如何把线性分数变成概率，并说清每个权重的作用。
> 2. 能写出二分类交叉熵，说明为什么错误且自信的预测会被重罚。
> 3. 能区分过拟合、欠拟合和正则化，并知道如何用验证集选择模型。
>
> **先修知识**：函数、求和符号、指数与对数、基础概率。
>
> **本章导读**：本章是后续矩阵分解、CTR 预估和深度模型的共同基础。公式不要求死记，但必须能解释输入、输出和参数含义。

2.1 模型到底在学什么



模型训练的基本闭环

推荐系统常见的机器学习形式包括：

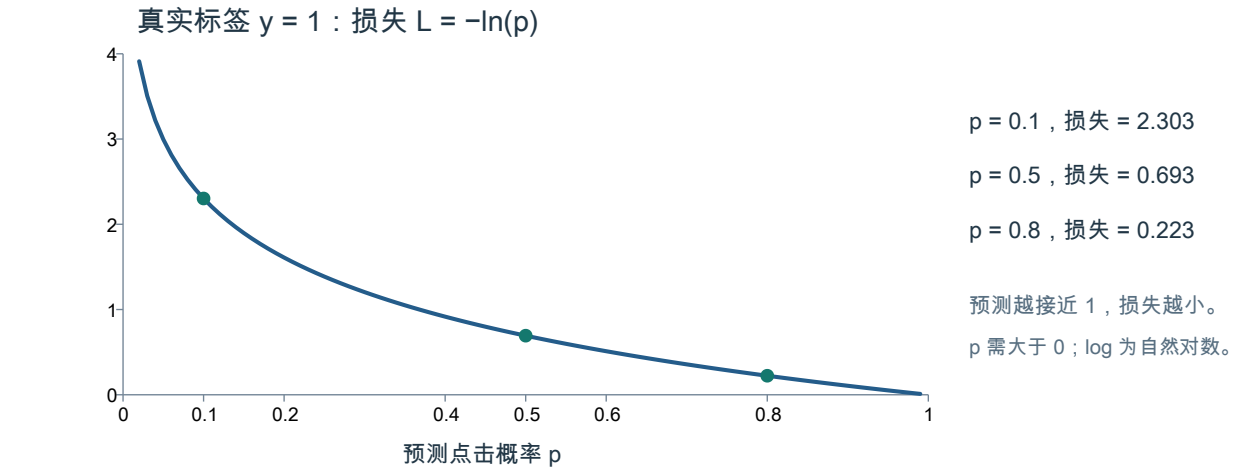
- **二分类**：是否点击、是否购买。
- **回归**：评分、观看时长、消费金额。
- **排序**：学习物品之间的相对先后关系。
- **表示学习**：学习用户和物品的向量表示，而不只追求一个预测分数。

训练的基本过程是：前向计算预测值 → 计算损失 → 反向传播梯度 → 更新参数 → 在验证集上选择模型和超参数。

可以把它想成练习投篮：模型先“投一次”（做预测），损失告诉它偏了多少，梯度提供调整方向，学习率决定每次调整多大。**参数**像练出来的手感，由训练更新；**超参数**像你事先定的练习规则，例如学习率、向量维度、练多少轮。

数据也有分工：训练集是练习题，验证集是用来决定练习方法的模拟考，测试集是最后的考试。看着最后考试的答案反复改模型，分数自然不能代表真实水平。

2.2 猜错以后，模型怎么改



真实标签为 1 时的交叉熵。横轴是预测概率 p，纵轴是单样本损失 −ln(p)。预测越接近真实标签，损失越小；不表示损失下降就保证线上收益。

逻辑回归将线性分数映射为概率：

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \qquad z = w_0 + \sum_j w_j x_j$$

公式参数说明

符号	含义	取值或维度
$\hat{y}$	模型预测的正类概率，即 $P(y = 1 \mid x)$	$(0, 1)$
z	线性组合得到的对数几率，也称为 logit	实数
$\sigma(z)$	Sigmoid 函数	输出 $(0, 1)$
w_0	截距或全局偏置	实数
w_j	第 j 个特征的权重	实数
x_j	样本的第 j 个特征值	实数；类别特征通常先编码
j	特征编号	$1, \ldots, d$，d 为特征数

公式在说什么

模型先把每个特征乘以自己的权重并求和，得到任意实数 z；Sigmoid 再把 z 压到 0 与 1 之间。z 越大，预测越接近 1；z 越小，预测越接近 0。若 $z = 0$，预测恰好为 0.5。该模型的对数几率是线性的：

$$\log \frac{\hat{y}}{1 - \hat{y}} = z$$

因此权重 w_j 可以理解为：在其他特征不变时，x_j 每增加 1 个单位，对数几率增加 w_j。这不是说概率直接增加 w_j。

二分类交叉熵为：

$$L = -\frac{1}{N} \sum_{n=1}^{N} \left[y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right]$$

公式参数说明

符号	含义	取值或维度
L	整个训练集的平均交叉熵损失	非负实数
N	参与计算的样本数量	正整数
n	样本编号	$1, \ldots, N$
y_n	第 n 个样本的真实标签	0 或 1
$\hat{y}_n$	模型对第 n 个样本预测的正类概率	$(0, 1)$
$\log$	自然对数	底数为 e

公式在说什么

当 $y_n = 1$ 时，$(1 - y_n) \log(1 - \hat{y}_n)$ 为 0，损失只剩 $-\log \hat{y}_n$；预测概率越接近 1，损失越小。当 $y_n = 0$ 时，损失只剩 $-\log(1 - \hat{y}_n)$；预测概率越接近 0，损失越小。前面的负号把“概率越小、惩罚越大”转换为非负损失。

数值直觉

- 真实标签为 1，预测 0.9：单样本损失约 0.105。
- 真实标签为 1，预测 0.5：单样本损失约 0.693。
- 真实标签为 1，预测 0.1：单样本损失约 2.303。

可见，错误且自信的预测会受到更大惩罚。

正则化用于约束模型复杂度。L2 正则倾向于让参数较小且平滑；L1 正则更容易产生稀疏参数。若只考虑权重 w_j，两种正则项可写为：

$$J_{L2} = L + \lambda \sum_j w_j^2, \qquad J_{L1} = L + \lambda \sum_j |w_j|$$

其中 J_{L2}、J_{L1} 分别表示加入 L2、L1 正则后的总训练目标，L 是原始训练损失，$\lambda \geq 0$ 控制正则强度，w_j 是第 j 个权重。λ 越大，模型越倾向于把权重压小或置零；$\lambda = 0$ 表示不加正则。截距 w_0 通常不参与正则化，因为它只负责整体平移，不会显著增加模型复杂度。

参数如何更新

训练模型时，通常用梯度下降或其变体更新参数：

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t)$$

公式参数说明

符号	含义	取值或维度
θ_t	第 t 次迭代时的全部模型参数	与模型同维的向量或参数集合
θ_{t+1}	更新后的参数	与 θ_t 同维
η	学习率，控制每次更新走多远	正实数，例如 0.001
$L(\theta_t)$	当前参数下的损失	非负实数
$\nabla_\theta L(\theta_t)$	损失对参数的梯度	与 θ_t 同维

梯度指向损失上升最快的方向，因此公式前面取负号，让参数沿下降方向移动。η 太大可能越过最优点甚至发散，太小则训练缓慢。实际系统常用 Mini-batch SGD、Adam 等变体，但基本思想仍是“根据梯度反复调整参数”。

泛化能力是模型对未见过数据的表现。训练损失下降而验证损失上升，通常意味着过拟合或训练、验证数据分布不一致。

2.3 先算一次预测，再认识神经网络积木

先做一个完整计算。设模型有截距 $w_0 = -1$，两个特征权重 $w_1 = 2$、$w_2 = -1$，某样本特征为 $x_1 = 1$、$x_2 = 0.5$，则：

$$z = -1 + 2 \times 1 + (-1) \times 0.5 = 0.5$$

$$\hat{y} = \sigma(0.5) = \frac{1}{1 + e^{-0.5}} \approx 0.622$$

这表示模型估计该样本属于正类的概率约为 62.2%。若真实标签为 1，则单样本交叉熵为：

$$-\log 0.622 \approx 0.475$$

若一次点击样本的 $y = 1$，模型预测 $\hat{y} = 0.8$，则该样本交叉熵为：

$$-\log 0.8 \approx 0.223$$

如果预测为 0.1，损失增至 $-\log 0.1 \approx 2.303$，说明模型对高置信度错误预测施加更大惩罚。

读深度模型前，再补一块积木

不必先把整本深度学习教材学完。先认识下面几个动作，就能读懂第 8～9 章的大部分结构：

- **Embedding：查表取向量。** 输入“电影 A 的编号”，取出一行可学习的数字。训练开始时，这些数字通常还没有偏好含义。

- **拼接：把几行信息接起来。** 用户向量有 8 个数，电影向量有 8 个数，拼在一起就有 16 个数；不是把对应位置相加。
- **全连接层：加权组合。** $Wx + b$ 是把输入重新混合成一组数，W 与 b 都可以学习。
- **激活函数：给组合加入非线性。** 例如 $ReLU(z) = \max(0, z)$。若多层之间始终只有线性变换，叠很多层仍等价于一个线性变换。
- **MLP：重复“全连接 + 激活”。** 最后接一层输出分数；若是二分类，可再用 Sigmoid 得到概率估计。
- **Softmax：在一组分数里分配份额。** $p_j = \exp(z_j) / \sum_k \exp(z_k)$，所有份额加起来为 1；其中 j, k 遍历这组候选。给这组候选加一个新选项，原有选项的份额也会变。
- **反向传播：沿计算过程把误差传回去。** 它计算每个参数的梯度，优化器再据此更新参数；不是每一步都人工修改网络。

例如“用户 ID → 用户向量”和“电影 ID → 电影向量”，把两个向量拼接后交给 MLP 输出分数，就是一个最小的深度打分模型。若分别编码两边，最后才做点积，则更接近双塔的思路。

2.4 常见误区与注意事项

- **训练集表现好不等于模型好。** 应重点观察验证集和按时间划分的测试集。
- **正负样本不平衡时不能只看准确率。** 全部预测为不点击也可能得到很高准确率。
- **分类损失与排序目标并不完全相同。** AUC 提升不一定意味着 NDCG@K 同步提升。
- **不要随意使用测试集调参。** 测试集应尽可能只用于最终评估。

本章小结

- 逻辑回归先计算线性分数 z，再用 Sigmoid 得到概率；权重影响的是对数几率，不是概率本身的绝对增量。
- 交叉熵只惩罚真实类别对应的预测概率，错误且高置信度的预测会产生更大损失。
- 正则化通过惩罚过大的参数抑制过拟合，λ 控制拟合训练数据与保持模型简单之间的权衡。
- 训练集用于学习参数，验证集用于选择模型和超参数，测试集应尽量只用于最终评估。

关键术语：Logit、Sigmoid、交叉熵、正则化、过拟合、泛化、验证集。

2.5 习题

1. 某模型训练损失持续下降，验证损失却开始上升，可能是什么原因？可采取哪些措施？
2. 为什么推荐系统中交叉熵优化良好，却仍可能得到不理想的推荐顺序？
3. L1 和 L2 正则化的典型差异是什么？

> **▼ 参考答案**
>
> 1. 可能是过拟合，也可能是训练与验证分布不同。可检查数据划分与泄漏，并尝试加强正则化、减少特征或模型复杂度、增加数据、早停等。
> 2. 交叉熵主要约束单个样本的概率预测，而推荐还关注同一列表内物品的相对顺序和前 K 个位置的质量。

3. L1 更容易产生稀疏解，L2 通常使参数整体收缩而较少直接变为零。

第 3 章　数据、样本构造与特征工程

本章学习目标

1. 能以一次曝光为单位构造训练样本，并明确标签的观察窗口。
2. 能区分用户、物品、上下文和交叉特征，理解 One-hot 与 Embedding 的取舍。
3. 能识别未来信息泄漏、曝光偏差和训练服务不一致。

先修知识：第 1～2 章；基础统计概念，如均值、比例和分布。

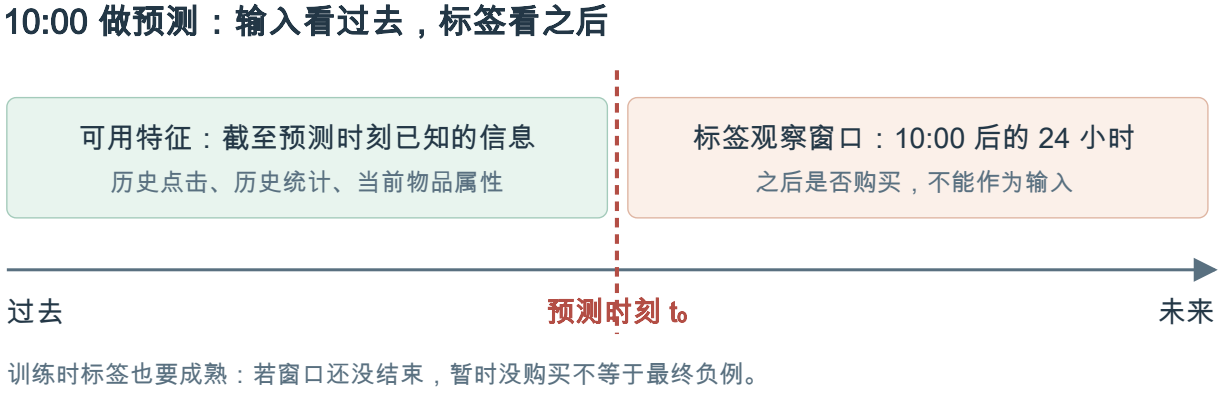
本章导读：推荐系统的上限很大程度由数据和特征决定。模型结构再复杂，也无法修复标签定义错误或未来信息泄漏。

3.1 一张日志表里，哪些信息能帮忙

推荐模型的数据通常来自日志。每条样本应尽量还原“请求发生时系统已知的信息”，常见特征包括：

- **用户特征**：年龄段、活跃度、历史偏好、近期行为统计。
- **物品特征**：类别、标签、作者、价格、热度、内容向量。
- **上下文特征**：时间、设备、网络、页面位置。
- **交叉特征**：用户类别偏好与物品类别的组合、用户与作者的历史交互。
- **序列特征**：最近点击或购买的物品序列及其时间间隔。

3.2 给每次预测拍一张当时的快照



先固定特征快照，再等待标签成熟。虚线是预测时刻。左侧信息可作输入，右侧结果用于事后打标签，不能反过来进入当前特征。

样本构造需要明确曝光、标签和观察窗口。把每行想成一张“当时的快照”：模型只能看快照里已经存在的信息，标签则在之后观察。

曝光时刻	用户	物品	曝光前近 7 天点击数	随后 1 小时内是否点击
10:00	小林	电影 A	8	1
10:05	小林	电影 B	9	0

这里一行是一次曝光，最后一列是标签，不是输入特征。1 小时只是教学示例，真实窗口按业务定义；10:05 那行能否使用第 9 次点击，要看它是否已在 10:05 前发生并可被系统读取。

转化任务还要定义归因规则：例如把哪次购买归给哪次曝光。若观察窗口是 24 小时，刚曝光 2 小时的样本还不能因为没买就记为最终负例；应等待窗口结束或单独处理延迟反馈。按时间切分后，还要检查训练标签在模型训练时是否已经可得。

类别编码：

- One-hot 简单但维度高，不能自动表达类别间相似性。
- Embedding 将类别映射为低维稠密向量，可通过训练学习语义和行为相似性。

数值处理：标准化、归一化、分桶和对数变换可缓解量纲、异常值和长尾分布问题。

数据泄漏指训练中使用了预测时不可得的信息，例如用“曝光后的点击次数”预测该次曝光是否点击，会得到虚高离线效果。

3.3 预测十点的点击，不能偷看十点以后的事

要预测用户是否在 10:00 点击商品：

- 合理特征：截至 09:59 的历史点击率、最近浏览类别、商品历史统计。
- 泄漏特征：包含 10:00 这次行为的全天点击次数、曝光后生成的用户标签。
- 更稳妥的统计方式：按时间窗口滚动计算，并确保统计窗口严格早于当前样本。

3.4 常见误区与注意事项

- **热门统计可能泄漏未来信息。** 应使用当前时刻之前的数据计算。
- **缺失值本身可能有意义。** 例如未填写年龄与随机缺失并不等价。
- **高基数 ID 容易过拟合。** 需要正则化、频次过滤、哈希或其他泛化机制。
- **训练与线上特征必须一致。** 同一特征若采用不同窗口或编码，会造成训练服务偏斜。
- **已曝光未点击也不是明确的“不喜欢”。** CTR 任务通常把观察窗口结束后“已曝光但未点击”记作标签 0；这是隐式反馈里的负标签，不是用户主动给的差评。记录曝光位置，并与召回任务中的“未交互采样负例”区分。

本章小结

- 一条推荐样本应还原“请求发生时系统已知的信息”，标签则在请求之后按明确窗口观察。

- 特征可按对象分为用户、物品、上下文和交叉信息，也可把历史行为单独组织为序列；类别特征常用 One-hot 或 Embedding，数值特征按模型需要处理。
- 数据泄漏的本质是训练使用了预测时不可得的信息，尤其是未来统计和曝光后行为。
- 训练与服务必须使用相同的特征定义、时间窗口和编码方式，否则离线提升无法在线复现。

关键术语：曝光、观察窗口、特征交叉、One-hot、Embedding、数据泄漏、训练服务偏斜。

3.5 习题

1. 用用户“未来 7 天的点击类别分布”预测其今天是否点击，为什么会造成泄漏？
2. 对长尾类别特征，Embedding 相比 One-hot 有什么优势和风险？
3. 为什么训练样本最好基于曝光日志，而不是把所有未交互物品都当作负样本？

> **▼ 参考答案**
>
> 1. 预测今天时无法知道未来 7 天行为，该特征使用了未来信息，会使离线结果虚高且线上不可复现。
> 2. 优势是低维稠密、可学习相似性并便于交叉；风险包括低频 ID 学习不足、参数规模大和哈希冲突等问题。
> 3. 所有未交互物品中包含大量未曝光物品，其标签并不等价于不喜欢；直接作为负样本会引入严重噪声和选择偏差。

第 4 章　离线评估、数据划分与在线指标

> **本章学习目标**
>
> 1. 能计算并解释 Precision@K、Recall@K、NDCG@K 和 AUC。
> 2. 能说明不同指标分别回答什么问题，以及为什么不能只看一个指标。
> 3. 能设计按时间划分的训练、验证和测试方案，并解释离线评估的局限。
>
> **先修知识**：第 2～3 章；集合、比例和排序的基本概念。
>
> **本章导读**：评估是推荐系统最容易出错的部分。指标算错、数据切错，后面的模型比较就全部失去意义。

4.1 给推荐器打分，先想清楚考什么

不同评估指标回答不同问题：

- LogLoss：概率预测是否准确，对错误置信度敏感。
- AUC：随机取一个正样本和一个负样本，正样本得分更高的概率。
- Precision@K：前 K 个推荐中有多少是相关的。

- **Recall@K**：用户所有相关物品中，有多少进入前 K。
- **NDCG@K**：相关性和位置共同决定的折损累计增益，越靠前权重越大。
- **覆盖率、多样性、新颖性**：衡量结果是否过度集中、是否重复、是否只推荐熟悉内容。

4.2 命中几个、排在哪里、概率准不准

先看最容易手算的两个指标。对用户 u，设推荐的前 K 个不同物品为 Rec_u^K，测试期可评估的相关物品集合为 R_u，则：

$$Precision_u@K = \frac{|Rec_u^K \cap R_u|}{K}, \qquad Recall_u@K = \frac{|Rec_u^K \cap R_u|}{|R_u|}$$

$\cap$ 表示交集，$|\cdot|$ 表示物品个数。想成“朋友列了 4 部想看的电影，你推荐 5 部，猜中 2 部”：Precision 是 $2/5$，Recall 是 $2/4$。这里的“相关”指数据里观察到、且符合事先标签规则的物品，不代表我们知道了用户全部真实喜好。

本指南默认先给每个有效用户算指标，再对用户取平均（宏平均）。R_u 为空的用户不参与这项 Recall/NDCG 平均，要报告排除人数。若候选不足 K，也要事先约定分母；下面的教学项目固定 Precision 分母为 K，不重复填充结果。

再往前一步：猜中了，放在哪里？ 两张片单都命中两部电影，把它们放在第 1、2 位通常比第 9、10 位更有用。DCG 就是在给靠前的命中更多分。一种常见定义为：

$$DCG@K = \sum_{j=1}^{K} \frac{2^{rel_j} - 1}{\log_2(j+1)}$$

公式参数说明

符号	含义	取值或说明
$DCG@K$	前 K 个结果的折损累计增益	非负实数，越大越好
K	只评估推荐列表的前多少个位置	正整数，例如 5 或 10
j	结果在列表中的位置，从 1 开始	$1, \dots, K$
rel_j	第 j 个结果的相关性等级	二值 0/1，或 0～4 等分级
$2^{rel_j} - 1$	第 j 个结果带来的增益	相关性越高，增益越大
$\log_2(j+1)$	位置折损分母	位置越靠后，分母越大，增益被压得越低

公式在说什么

排名第 1 的结果，分母是 $\log_2(2) = 1$，没有位置折损；排名越靠后，分母越大，即使相关性相同，贡献也越小。若某位置的结果不相关，$rel_j = 0$，则 $2^0 - 1 = 0$，该位置不贡献增益。

再以理想排序的 $IDCG@K$ 归一化：

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

公式参数说明

符号	含义	取值或说明
$NDCG@K$	归一化折损累计增益	通常为 $[0,1]$，越大越好
$DCG@K$	当前推荐列表的折损累计增益	由上一个公式计算
$IDCG@K$	理想排序下的最大 DCG	在事先确定的评估范围内，将全部已知相关物品按等级从高到低排列后取前 K；不是只重排已命中的物品

公式在说什么

IDCG 是“如果系统把最相关的结果全部排在最前面，最多能拿多少分”。用 DCG 除以 IDCG，就把不同用户、不同数量相关物品的得分归一到可比较的尺度。若前 K 位达到该评估范围内的理想相关性顺序，NDCG 为 1；排序越差，NDCG 越接近 0。若某用户没有任何相关物品，IDCG 可能为 0，此时应跳过该用户或预先定义特殊值，不能直接做除法。

AUC 的精确定义

AUC 不是“准确率”，而是正样本得分高于负样本的概率。设正样本集合为 $\mathcal{P}$，负样本集合为 $\mathcal{N}$，则：

$$AUC = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{p\in\mathcal{P}} \sum_{n\in\mathcal{N}} \left(\mathbf{1}[s_p > s_n] + \frac{1}{2}\mathbf{1}[s_p = s_n] \right)$$

公式参数说明

符号	含义	取值或说明
AUC	正样本排在负样本之前的概率	$[0,1]$，0.5 约等于随机排序
$\mathcal{P}$	所有正样本的集合	例如所有真实点击样本
$\mathcal{N}$	所有负样本的集合	例如所有真实未点击样本
p, n	正样本和负样本的编号	$p \in \mathcal{P}$，$n \in \mathcal{N}$
s_p, s_n	模型给正、负样本的分数	同一模型下的可比实数
$\mathbf{1}[\cdot]$	指示函数，条件成立为 1，否则为 0	0 或 1
$\lvert\mathcal{P}\rvert, \lvert\mathcal{N}\rvert$	正、负样本数量	正整数

将所有正负样本两两配对，若正样本得分更高记 1 分，相同记 0.5 分，更低记 0 分，最后求平均。因此 AUC 只关心相对顺序，不关心概率数值本身是否校准。给所有原始排序分数加上 10，AUC 不变；对分数做严格递增变换也不改变排序。不过，概率本身必须在 $[0,1]$ 内，不能把加过 10 的数再当 CTR。若评估集合只有正样本或只有负样本，AUC 无定义，不应硬填成 0.5。

数据划分应尽量模拟真实上线：

- 用较早时间段训练，较晚时间段验证和测试。
- 防止同一事件派生出的样本同时进入训练和测试。

- 评估召回时，需要明确相关物品集合以及候选生成方式。全库评估和“1 个正例 + 99 个采样负例”是两套不同难度的考试，分数不能混比。
- 时间切分更适合检验“用过去预测未来”。随机切分也有用途，例如独立同分布的评分补全实验；关键是说清它回答什么问题，不能把它冒充未来上线表现。
- 仅按用户分别保留最后一次行为，也不保证全局时间一致：甲的训练行为可能晚于乙的测试行为。模拟固定时点上线时，优先使用统一时间边界。

4.3 手算一张片单的成绩

同样命中两部，靠前的命中获得更高 NDCG

片单 A



Precision@5 = 2/5 = 0.4

Recall@5 = 2/4 = 0.5

NDCG@5 = 0.637

片单 B



Precision@5 = 2/5 = 0.4

Recall@5 = 2/4 = 0.5

NDCG@5 = 0.319

设评估范围有 4 个二值相关物品。IDCG 参考这 4 个物品，不只重排已命中的 2 个。

同命中，不同排序分。两张片单的 Precision 和 Recall 相同；A 的相关结果更靠前，因此 NDCG 更高。数字由二值相关性、K=5、4 个相关物品计算。

某用户测试集中共有 4 个相关物品，推荐前 5 个结果命中其中 2 个。记：

- $K = 5$：只检查前 5 个位置；
- $H = 2$：前 5 个位置中命中相关物品的数量；
- $R = 4$：测试集中相关物品的总数。

则：

$$Recall@5 = \frac{H}{R} = \frac{2}{4} = 0.5, \qquad Precision@5 = \frac{H}{K} = \frac{2}{5} = 0.4$$

Recall 的分母是“所有相关物品”，回答“相关内容找回了多少”；Precision 的分母是“展示出来的 K 个位置”，回答“展示出来的内容有多少是相关的”。两者的分母不同，不能混用。

再看位置的影响。假设另一个用户只有 2 个相关物品，系统把二者都命中，但分别排在第 3、4 位。按二值相关性 $rel_j \in \{0, 1\}$ 计算：

$$DCG@5 = \frac{1}{\log_2(4)} + \frac{1}{\log_2(5)} \approx 0.5000 + 0.4307 = 0.9307$$

理想排序应把它们放在第 1、2 位：

$$IDCG@5 = \frac{1}{\log_2(2)} + \frac{1}{\log_2(3)} \approx 1 + 0.6309 = 1.6309$$

所以：

$$NDCG@5 = \frac{0.9307}{1.6309} \approx 0.571$$

若同样两个结果被移到第 1、2 位，则 $DCG@5 = IDCG@5$，NDCG 提升为 1。这个例子说明：Recall 和 Precision 只看“有没有命中”，NDCG 还会惩罚“命中了但排得太后”。

4.4 常见误区与注意事项

- **AUC 高不代表头部排序好。** 全局两两区分能力强，不代表前 K 个结果优秀。
- **随机切分可能过于乐观。** 用户未来行为容易泄漏到训练集。
- **离线相关标签不等于真实满意度。** 点击、购买和长期留存代表不同目标。
- **指标需与环节对应。** 召回通常更关注 Recall@K；精排还会关注 AUC、LogLoss 或 NDCG。
- **注意显著性与分层。** 不同用户活跃度、物品热度下的平均表现可能掩盖局部退化。
- **GAUC 需要写清规则。** 它通常是先按用户或请求分组算 AUC，再按样本数或曝光数等加权。只有一种标签的组无法算 AUC，应排除并报告数量。不同分组、权重得到的 GAUC 不能直接比较。

本章小结

- Precision@K 的分母是展示位置数 K，Recall@K 的分母是相关物品总数；两者回答的问题不同。
- NDCG 同时考虑相关性和位置，越靠前的结果权重越高；AUC 只关注正负样本的相对顺序。
- AUC 不是准确率，也不代表概率校准；排序好不代表预测概率接近真实发生率。
- 推荐评估应采用符合线上使用方式的时间切分，并明确候选集合、相关标签和已见过滤规则。

关键术语：Precision@K、Recall@K、NDCG@K、AUC、时间切分、候选集合。

4.5 习题

1. 某用户有 10 个相关物品，前 10 个推荐命中 3 个，求 Recall@10 和 Precision@10。
2. 为什么两个模型 Recall@10 相同，NDCG@10 却可能不同？
3. 为什么推荐系统常采用按时间划分而不是纯随机划分？

▼ **参考答案**

1. Recall@10 = 3/10 = 0.3；Precision@10 = 3/10 = 0.3。二者数值恰好相同，但含义不同。
2. 命中物品的位置可能不同；NDCG 对靠前位置赋予更高权重。
3. 按时间划分更接近用历史预测未来的真实使用方式，可减少未来信息泄漏和过于乐观的评估。

第 5 章　基线、内容推荐与协同过滤

> **本章学习目标**
>
> 1. 能说明基线模型的作用，以及为什么复杂模型必须与热门推荐比较。
> 2. 能区分基于内容的推荐与协同过滤，理解 UserCF、ItemCF 的不同视角。
> 3. 能计算余弦相似度，并解释热门物品和冷启动对协同过滤的影响。
>
> **先修知识**：第 3～4 章；向量、点积和余弦的基本概念。
>
> **本章导读**：本章从最简单的可运行方法开始，建立“先有可靠基线，再有复杂模型”的工程习惯。

5.1 先问热门榜，再问相似的人和物

基线模型用于判断复杂模型是否真正带来价值，常见基线包括热门推荐、最近行为推荐和简单规则。

基于内容的推荐利用物品属性和用户历史偏好，例如根据用户喜欢的电影类型、导演和演员推荐相似电影。

协同过滤利用群体交互行为，不要求先理解物品内容：

- UserCF：找到相似用户，推荐他们喜欢而当前用户未接触的物品。
- ItemCF：根据物品共现或相似行为，推荐与历史物品相似的物品。

5.2 用共同观众给相似度投票

用户向量可由其交互物品构成，余弦相似度为：

$$sim(u,v)=\frac{r_u\cdot r_v}{\|r_u\|\|r_v\|}$$

公式参数说明

符号	含义	取值或维度
$sim(u,v)$	用户 u 与用户 v 的余弦相似度	$[-1,1]$；非负向量时为 $[0,1]$
r_u	用户 u 的交互向量，每个维度对应一个物品	长度等于物品数
r_v	用户 v 的交互向量	与 r_u 同维度
$r_u\cdot r_v$	两个向量的点积，对应维度相乘后求和	实数
$\|r_u\|$	向量长度 $\sqrt{\sum_i r_{u,i}^2}$	非负实数
$r_{u,i}$	用户 u 对物品 i 的交互值	0/1，或评分、点击次数等

公式在说什么

余弦相似度比较的是两个交互向量的方向，而不是长度。若 $r_u=[1,0,1]$，$r_v=[1,1,0]$，则：

$$sim(u,v)=\frac{1\times1+0\times1+1\times0}{\sqrt{2}\sqrt{2}}=\frac{1}{2}=0.5$$

两人只共同交互了第一个物品，因此相似度为正但不高。若把所有物品都交互过的用户与只交互一个物品的用户比较，原始点积会被活跃用户的数量优势抬高；余弦归一化可以部分缓解这种长度影响，但仍不能完全消除热门物品和曝光偏差。

ItemCF 像书店里“买过这本书的人，也买过……”的书架：相似来自读者行为，不一定来自书的主题。实际系统通常要做归一化或热门降权，否则畅销书会在几乎每个书架上出现。

对二值交互，设 U_i 为训练期与物品 i 交互过的用户集合，则物品余弦相似度可写为：

$$sim(i,j) = \frac{|U_i \cap U_j|}{\sqrt{|U_i||U_j|}}, \qquad score(u,j) = \sum_{i \in H_u} sim(i,j)$$

H_u 是用户训练期的交互集合。第一式数“共同读者”并归一化；第二式让用户读过的每本书给候选书投一票。这里只演示最简单的等权求和，实际可截断邻居、按时间衰减。空交互向量的余弦没有定义，新用户或新物品应走热门、内容等兜底。

5.3 动手：只用 Python，给小林挑两部电影

下面用 A ~ E 代替电影名。只需 Python 3，不需要下载数据、安装框架或使用 GPU。把代码保存为 `itemcf_demo.py`，在终端运行 `python3 itemcf_demo.py`，也可以直接在 Notebook 里执行。

```
from math import sqrt

# 假设这是某个时间边界以前的全部交互。set 自动去重。
train = {
    "小林": {"A", "B"},
    "小王": {"A", "C"},
    "小张": {"B", "C", "D"},
    "小李": {"A", "D", "E"},
}

# 倒过来建表：每部电影有哪些观众。只读训练数据。
item_users = {}
for user, items in train.items():
    for item in items:
        item_users.setdefault(item, set()).add(user)


def similarity(a, b):
    ua, ub = item_users[a], item_users[b]
    return len(ua & ub) / sqrt(len(ua) * len(ub))


seen = train["小林"]
candidates = set(item_users) - seen  # 看过的 A、B 不再推荐
scores = {
    item: sum(similarity(old, item) for old in sorted(seen))
    for item in candidates
}
# 分数相同时按电影名排序，避免每次结果不同。
ranked = sorted(candidates, key=lambda item: (-scores[item], item))
for item in ranked:
```

```
    print(item, round(scores[item], 3))

# 假设之后观察到小林喜欢 D；这里只用于评估，不参与计算相似度。
relevant = {"D"}
k = 2
top_k = ranked[:k]
hits = len(set(top_k) & relevant)
print("推荐：", top_k)
print(f"Precision@{k} = {hits / k:.2f}")
print(f"Recall@{k} = {hits / len(relevant):.2f}")
```

预期输出：

```
C 0.908
D 0.908
E 0.577
推荐： ['C', 'D']
Precision@2 = 0.50
Recall@2 = 1.00
```

为什么 C 靠前？C 与 A 有共同观众小王，与 B 有共同观众小张，两部分相似度相加约为 $0.408 + 0.500 = 0.908$。D 也拿到同样的票数，按事先约定用名称打破平局。E 只与 A 有共同观众，得分约 0.577。

试一试：先预测“如果小李也看过 B，E 的得分会怎样变化”，再改训练数据验证。解释得通，比只看到程序运行成功更重要。这里的数据是人为构造的；一个用户的 Recall=1 只用于检查算法与指标，不能证明它能在真实用户上取得好效果。

5.4 常见误区与注意事项

- **相似不等于因果。** 共现可能由活动、热度或曝光位置造成。
- **UserCF 与 ItemCF 各有适用条件。** 用户规模、物品规模和关系稳定性会影响选择。
- **协同过滤存在冷启动。** 新用户、新物品缺乏行为时难以计算相似性。
- **热门推荐并不总是弱基线。** 在数据极少或需求高度集中时，它可能非常有效。
- **内容特征也可能产生偏置。** 标签错误和粒度不统一会直接影响结果。

本章小结

- 基线模型提供效果下限，热门推荐、最近行为推荐和简单规则应始终保留作为对照。
- 内容推荐依赖物品属性，适合冷启动；协同过滤依赖群体行为，不要求理解内容。
- UserCF 从“相似用户”出发，ItemCF 从“相似物品”出发；余弦相似度通过归一化部分缓解活跃度差异。
- 共现相似不等于因果偏好，热门物品、曝光位置和活动运营都可能制造虚假关系。

关键术语：基线、基于内容的推荐、UserCF、ItemCF、余弦相似度、冷启动。

5.5 习题

1. 片库今天新增一部电影，有类型、简介和演员信息，却还没有观众评分。为什么内容推荐已经能用，ItemCF 却难办？
2. 为什么计算物品共现相似度时通常要处理热门物品？
3. 在构建复杂深度模型前，为什么应先保留热门或 ItemCF 基线？

> ▼ **参考答案**
>
> 1. 内容推荐可使用类别、文本等已有属性；ItemCF 需要与其他物品的行为共现，新物品通常没有足够交互。
> 2. 热门物品会与大量物品共现，若不降权，可能产生缺乏区分度的相似关系。
> 3. 基线提供可比较的效果下限，有助于判断提升是否来自模型本身，并便于定位数据或评估问题。

第 6 章　矩阵分解与隐式反馈推荐

> **本章学习目标**
>
> 1. 能解释矩阵分解中全局均值、用户偏置、物品偏置和隐向量内积各自的作用。
> 2. 能读懂带正则化的平方损失，并说明 Ω 与 λ 的含义。
> 3. 能区分显式评分与隐式反馈，理解 BPR 为什么优化相对顺序而非绝对评分。
>
> **先修知识**：第 2 章损失函数；第 5 章协同过滤；向量内积与 L2 范数。
>
> **本章导读**：矩阵分解是从传统推荐走向 Embedding 和深度模型的桥梁。请重点理解“向量维度必须一致”和“分数不是概率”这两个细节。

6.1 给用户和电影各做一张数字名片

上一章靠“共同看过什么”找相似电影。这一章换个办法：给每个用户和每部电影各做一张数字名片，用几列数字概括匹配关系。两张名片一比较，就能算出偏好分数，不必把所有用户与电影的关系都存满。

这就是矩阵分解的一种理解方式：把用户和物品映射到同一潜在向量空间，用内积打分。名片上的列由训练学出，通常不能直接命名为“科幻喜好”“喜剧程度”；那只是帮助入门的类比。

常见预测形式为：

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$$

公式参数说明

符号	含义	取值或维度
$\hat{r}_{ui}$	模型预测的用户 u 对物品 i 的评分或偏好分数	实数；若用评分数据训练，单位与评分相同
μ	全局平均评分，吸收整体水平	实数，例如全部评分的均值 3.5
b_u	用户偏差，表示用户整体打分偏高或偏低	实数，可正可负
b_i	物品偏差，表示物品整体受欢迎或不受欢迎的程度	实数，可正可负
p_u	用户 u 的隐向量	d 维列向量
q_i	物品 i 的隐向量	d 维列向量
$p_u^T q_i$	用户向量与物品向量的内积，表示兴趣匹配程度	实数
d	隐向量维度，是人工设置的超参数	正整数，例如 32、64

公式在说什么

可以把预测评分拆成四块：大家平均会给多少分（μ），这个用户比平均更宽松还是更严格（b_u），这个物品比平均更受欢迎还是更冷门（b_i），以及这个用户与这个物品的潜在兴趣是否匹配（$p_u^T q_i$）。内积展开为：

$$p_u^T q_i = \sum_{f=1}^{d} p_{u,f} q_{i,f}$$

其中 $p_{u,f}$ 和 $q_{i,f}$ 分别是用户向量和物品向量在第 f 个潜在维度上的取值。维度本身通常没有可命名的业务含义，模型只要求同一维度上的数值能够共同解释偏好。两个向量必须维度相同，才能逐维相乘并求和。

数值例子

若 $\mu = 3.5$，$b_u = 0.3$，$b_i = -0.2$，$p_u^T q_i = 0.4$，则：

$$\hat{r}_{ui} = 3.5 + 0.3 - 0.2 + 0.4 = 4.0$$

这表示模型预测该用户会给该物品约 4 分。这里的 4.0 是评分尺度下的预测，不是概率。

6.2 评分要猜得准，排序要分出先后

显式评分场景可对已观察评分最小化：

$$L = \sum_{(u,i)\in\Omega} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 + \sum_u b_u^2 + \sum_i b_i^2 \right)$$

公式参数说明

符号	含义	取值或说明
L	训练目标值，越小表示拟合越好且参数越受约束	非负实数
Ω	训练集中所有已观察的用户—物品评分对	集合

符号	含义	取值或说明
(u, i)	一个已观察的评分样本	u 为用户，i 为物品
r_{ui}	真实评分	例如 1～5
$\hat{r}_{ui}$	模型预测评分	实数
$(r_{ui} - \hat{r}_{ui})^2$	单样本平方误差，惩罚预测偏离	非负实数
λ	正则化强度	非负实数；越大，参数越受压制
$\sum_u$	对所有用户的对应参数求和	包括训练中出现的用户
$\sum_i$	对所有物品的对应参数求和	包括训练中出现的物品
$\|p_u\|^2$	用户向量各分量平方和	非负实数
$\|q_i\|^2$	物品向量各分量平方和	非负实数
b_u^2, b_i^2	对用户、物品偏差的平方惩罚	非负实数

公式在说什么

第一项让预测评分尽量接近真实评分，只对已经观察到的评分对 Ω 求和；第二项对所有用户向量、物品向量和偏置做平方惩罚，防止参数变得过大。若只拟合已有评分而不加正则，模型可能给少数活跃用户和热门物品分配很大的向量，从而记住训练噪声。λ 控制“拟合训练数据”和“保持参数平滑”之间的权衡。

隐式反馈中，“未交互”不是可靠负评分。常用思路包括：

- 为不同交互设置置信度，例如购买比单次点击更强。
- 使用采样构造训练对。
- 使用 BPR 等成对排序目标，让已交互物品得分高于采样负物品。

BPR 对三元组 (u, i, j) 优化：

$$L_{BPR} = -\log \sigma(s_{ui} - s_{uj}) + \lambda \|\theta\|^2$$

公式参数说明

符号	含义	取值或说明
L_{BPR}	一个三元组上的贝叶斯个性化排序损失	实数，越小越好
u	当前用户	用户编号
i	用户曾经交互过的正物品	物品编号
j	为用户采样、通常未交互的负物品	物品编号，$j \neq i$
s_{ui}	模型给用户 u 与正物品 i 的分数	实数
s_{uj}	模型给用户 u 与负物品 j 的分数	实数
$s_{ui} - s_{uj}$	正物品相对负物品的分数差	实数
$\sigma(\cdot)$	Sigmoid 函数	把分数差映射到 $(0, 1)$
λ	正则化强度	非负实数
θ	模型全部可学习参数	包括用户、物品向量等

符号	含义	取值或说明
$\|\|\theta\|\|^2$	所有参数平方和	非负实数

公式在说什么

BPR 不要求模型预测“点击概率为多少”，而要求正物品的分数高于负物品。若 s_{ui} 比 s_{uj} 大很多，$\sigma(s_{ui} - s_{uj})$ 接近 1，$-\log(\cdot)$ 接近 0；若排序反了，Sigmoid 变小，损失迅速增大。正则项抑制参数无限增大。

数值直觉

若 $s_{ui} = 1.2$、$s_{uj} = 0.3$，分数差为 0.9：

$$-\log\sigma(0.9) \approx -\log(0.711) \approx 0.341$$

若两者顺序反转，$s_{ui} - s_{uj} = -0.9$：

$$-\log\sigma(-0.9) \approx -\log(0.289) \approx 1.241$$

同一对物品，排反后损失约变为原来的 3.6 倍，因此模型会被推向正确顺序。

6.3 两张名片怎样算出一个分数

若 $p_u = [1, 2]$，$q_i = [2, 0.5]$，忽略偏差项，则：

$$\hat{r}_{ui} = 1 \times 2 + 2 \times 0.5 = 3$$

该数值是偏好分数，不应未经额外建模就解释为点击概率。

6.4 常见误区与注意事项

- **内积分数不是天然概率。** 如需概率，应配合合适的输出函数和训练目标。
- **隐式反馈不能简单将所有缺失项置为 0 分。** 这会混淆未曝光与不喜欢。
- **负采样分布会改变模型行为。** 过于简单可能学不到难负例，过于困难又可能引入假负例。
- **向量维度不是越大越好。** 更大维度增加表达力，也增加过拟合和计算成本。

本章小结

- 矩阵分解用“全局均值 + 用户偏置 + 物品偏置 + 隐向量内积”预测偏好；用户向量与物品向量必须同维。
- 显式评分使用带正则化的平方损失，Ω 只包含已观察评分，λ 控制参数惩罚强度。
- 隐式反馈中，未交互不等于负样本；BPR 通过“正物品分数高于采样负物品”学习相对顺序。
- 矩阵分解输出的是偏好分数，不是概率；若要用于 CTR，还需概率变换和校准。

关键术语：矩阵分解、隐向量、全局均值、用户偏置、物品偏置、BPR、负采样。

6.5 习题

1. 矩阵分解中的用户向量和物品向量为什么必须处于同一维度？

2. BPR 的直接优化目标是什么？它与逐点评分误差有何区别？
3. 将交互次数多的样本设置更高置信度，有什么直觉依据？

> **▼ 参考答案**
>
> 1. 需要通过逐维相乘求内积得到偏好分数，因此向量维度必须一致。
> 2. 让同一用户的正物品得分高于负物品，优化相对顺序；逐点方法主要拟合每个交互的独立数值或概率。
> 3. 重复或更强行为通常比偶然单次行为更能稳定反映偏好，但具体权重仍需验证。

第 7 章　CTR 预估：逻辑回归、FM 与树模型

> **本章学习目标**
>
> 1. 能准确解释 CTR 的条件概率含义，知道它与“喜欢程度”的区别。
> 2. 能说明逻辑回归、FM 和 GBDT 在特征关系建模上的差异。
> 3. 能读懂 FM 的线性项与二阶交互项，并理解稀疏特征为何需要隐向量。
>
> **先修知识**：第 2 章逻辑回归与交叉熵；第 3 章特征工程；第 4 章 AUC 与 LogLoss。
>
> **本章导读**：本章进入工业推荐最核心的排序问题。学习时不要只记模型缩写，而要明确每个模型新增了哪种表达能力。

7.1 预测会不会点开，不等于预测喜不喜欢

CTR（Click-Through Rate）预估以曝光为样本，预测点击概率：

$$P(y=1 \mid u, i, c)$$

公式参数说明

符号	含义	取值或说明
$P(y=1 \mid u, i, c)$	在用户、物品和上下文已知时，发生正行为（通常为点击）的条件概率	$[0, 1]$
y	行为结果随机变量	1 表示点击或转化，0 表示未发生
u	当前用户	用户编号或特征集合
i	当前曝光的物品	物品编号或特征集合
c	当前上下文，例如时间、设备、位置	可为类别或数值特征

公式在说什么

CTR 预估不是预测“某个用户一定点击”，而是在给定这次曝光条件时，估计正行为发生的概率。条件竖线“|”表示“在已知 u, i, c 的前提下”。训练标签来自曝光日志：曝光后发生点击记为 1，未点击记为 0；但未点击还可能是位置、审美或需求等原因，因此 CTR 只是可观测行为，不是完整偏好。

它是广告、内容推荐和搜索排序中的常见模块，但点击只是目标之一，并不必然等价于用户满意度。

7.2 单个特征不够，还要看它们怎么搭配

逻辑回归训练快、可解释性较强，但主要表达线性关系，需要人工构造有效交叉特征。

FM（Factorization Machines）通过隐向量内积学习二阶特征交互：

$$\hat{y} = w_0 + \sum_j w_j x_j + \sum_{j<k} \langle v_j, v_k \rangle x_j x_k$$

公式参数说明

符号	含义	取值或维度
$\hat{y}$	FM 的原始预测分数；用于 CTR 时通常再经过 Sigmoid 得到概率	实数
w_0	全局偏置	实数
j, k	特征编号	$1, \ldots, d$，d 为特征总数
x_j, x_k	第 j、第 k 个特征的取值	实数；One-hot 场景下常为 0 或 1
w_j	第 j 个特征的线性权重	实数
v_j	第 j 个特征的隐向量	m 维列向量
v_k	第 k 个特征的隐向量	与 v_j 同为 m 维
$\langle v_j, v_k \rangle$	两个隐向量的内积 $v_j^T v_k$，表示特征交互强度	实数
m	隐向量维度，是超参数	正整数
$\sum_{j<k}$	对所有不重复的特征对求和	每对只计算一次

公式在说什么

第一项 w_0 是全局基线；第二项给每个特征一个独立线性贡献；第三项让特征两两交互。每个特征不只拥有一个权重，还拥有一个隐向量，两个特征是否“合拍”由隐向量内积决定。这样即使特征 j 和 k 在训练集中从未共同出现，只要它们各自与其他特征发生过交互，隐向量仍可能被学习，从而对未见组合做出估计。

数值例子

设 $x_j = 1$、$x_k = 1$，$v_j = [1, 0.5]$，$v_k = [0.5, 1]$，则：

$$\langle v_j, v_k \rangle = 1 \times 0.5 + 0.5 \times 1 = 1.0$$

这一对特征对分数的贡献为 $1.0 \times 1 \times 1 = 1.0$。若 $x_k = 0$，说明该特征未出现，交互项贡献为 0。

注意：标准 FM 主要建模二阶交互，不自动包含所有三阶、四阶组合；它也不是“训练后自动解释每个隐维度”的模型。

GBDT/XGBoost/LightGBM通过多棵树的加性组合拟合非线性关系，对表格特征和统计特征通常有较强表现。树模型也可与 LR 或深度模型组合。

7.3 “爱看科幻”遇上“一部科幻片”

“爱看科幻”和“这是一部科幻片”各自都提供一点信息，但它们**碰在一起**才更能说明这次匹配。LR 通常需要你提前写出这个组合特征；FM 让两个特征的向量“碰一碰”；树模型则可以沿“先问偏好，再问电影类型”的分裂路径表达组合关系。

再看采样：原本 1000 次曝光里有 10 次点击，点击率是 1%。若保留全部点击，却只随机保留约十分之一未点击样本，训练集就变成约 10 次点击、99 次未点击，正例占比约 9.2%。模型在这样的数据上输出的概率，自然不能直接当作原曝光分布的 CTR。

这与 AUC 是两回事。若在正、负类内部各自独立随机抽样，理想总体 AUC 不因类别比例改变而改变，不过有限样本估计会波动；若专挑难负例、只留部分人群，类内分布也变了，AUC 就可能明显变化。

7.4 常见误区与注意事项

- **FM 并不自动解决所有高阶交互。** 标准 FM 主要建模二阶交互。
- **CTR 高可能伴随低质量。** 还需观察转化、留存、投诉和长期指标。
- **类别不平衡不等于必然失准。** 改变类别比例的采样或类别加权，可能让输出偏离原分布；应检查校准，不能把“少数类少”本身当作失准证明。
- **树模型不天然适合所有超高维稀疏 ID。** 需要结合编码、统计特征或 Embedding 方法。
- **特征重要性不等于因果效应。** 高重要性特征可能只是与其他因素相关。

本章小结

- CTR 是在给定用户、物品和上下文的条件下，曝光后发生点击的条件概率。
- 逻辑回归表达线性对数几率，需要人工构造交叉特征；FM 用特征隐向量内积自动学习二阶交互；树模型擅长拟合表格特征的非线性关系。
- FM 的每个特征同时拥有线性权重和隐向量，隐向量维度是超参数，标准 FM 主要建模二阶交互。
- CTR 高不等于用户满意；负采样后模型输出可能失去真实概率含义，需要重新校准。

关键术语：CTR、Logit、特征交叉、FM、GBDT、概率校准、负采样。

7.5 习题

1. 相比 LR，FM 为什么更适合稀疏特征交互？
2. 负采样后 CTR 模型的 AUC 仍较好，但预测概率整体偏高或偏低，可能是什么原因？
3. 为什么不能只凭 CTR 决定最终推荐列表？

▼ **参考答案**

1. FM 用特征隐向量的内积表示交互，可共享参数并泛化到稀疏特征组合；普通 LR 的交叉项更依赖显式构造和直接观察。
2. 采样改变了训练数据中的正负比例，若不做概率校正或重新校准，输出不再对应原始分布下的 CTR。
3. 还需考虑转化、内容质量、多样性、长期满意度和业务约束；单纯优化 CTR 可能鼓励诱导点击。

第 8 章　Embedding、双塔召回与向量检索

本章学习目标

1. 能解释 Embedding 的作用，以及双塔模型为什么适合大规模召回。
2. 能区分点积与余弦相似度，并说明训练和检索时必须保持一致。
3. 能识别随机负样本、批次内负样本和难负样本各自的风险。

先修知识：第 6 章隐向量；第 7 章 CTR 排序；基础神经网络与向量运算。

本章导读：本章的关键不是“双塔”这个名称，而是“用户和物品分别编码、物品向量可预计算、线上用 ANN 检索”这一工程闭环。

8.1 先各自做名片，最后再比较

Embedding 是将用户、物品或特征映射为稠密向量的表示方法。可以把 Embedding 表理解成一张可学习的查找表：每个离散 ID 对应一行向量，训练时通过反向传播不断调整这些行。One-hot 像给每个 ID 发一个独立编号：两个不同 ID 的向量内积为 0，表达不出“它们有多像”。Embedding 则像给它们放进一张可学习的坐标图。哪些点会靠近取决于训练任务与损失，不保证“经常共同出现就一定相近”。

在召回阶段，常用**双塔模型**分别编码用户和物品：

$$e_u = f_\theta(x_u), \qquad e_i = g_\phi(x_i)$$

公式参数说明

符号	含义	取值或维度
e_u	用户塔输出的用户表示向量	d 维列向量
e_i	物品塔输出的物品表示向量	与 e_u 同为 d 维
x_u	用户侧原始特征，例如用户 ID、年龄、历史统计	维度由用户特征决定
x_i	物品侧原始特征，例如物品 ID、类目、内容向量	维度由物品特征决定
f_θ	用户塔函数，例如 Embedding 加 MLP	参数为 θ
g_ϕ	物品塔函数，例如 Embedding 加 MLP	参数为 ϕ

符号	含义	取值或维度
θ, ϕ	两个塔各自的可学习参数	两组参数，通常不共享
d	输出向量维度，是超参数	正整数，例如 64、128

公式在说什么

用户塔把用户特征压缩成一个向量，物品塔把物品特征压缩成另一个向量。两塔最后输出必须维度相同，才能计算内积：

$$s(u,i) = e_u^T e_i$$

其中 $s(u,i)$ 是用户与物品的匹配分数。若使用余弦相似度，则还需除以两个向量的长度：

$$\cos(e_u, e_i) = \frac{e_u^T e_i}{\|e_u\| \|e_i\|}$$

点积同时受方向和长度影响；余弦只关心方向。选择哪一种，要与训练目标一致，不能在训练时用点积、线上检索时随意改成余弦。

8.2 物品名片提前做好，请求来了就快找

为什么适合召回

物品塔只依赖物品特征，因此物品向量可以在离线阶段全部算好并建立索引。线上来了一个用户请求，只需计算一次用户向量，再用近似最近邻检索找最相似的物品。这样就不必对全库每个物品运行复杂模型。

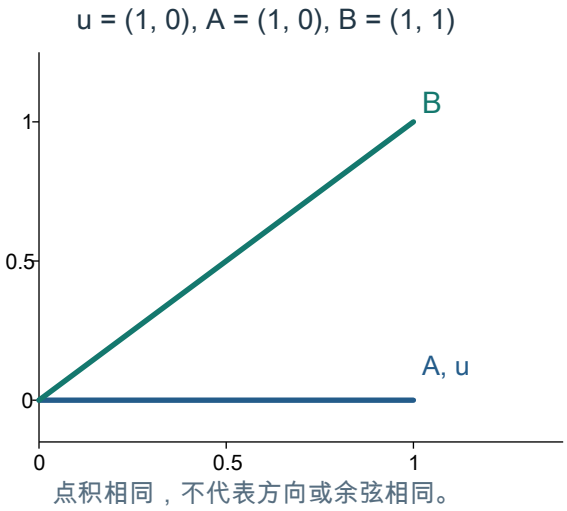
训练常使用 Softmax 或对比学习目标，让正物品相对负物品的匹配分数更高。只有在特定归一化和距离定义下，才可直接把这解释成几何上的“拉近、推远”。常见负样本来源：

- 随机负样本。
- 批次内其他正样本，即 in-batch negatives。
- 难负样本，例如与正样本相似但未交互的物品。

小数据先做精确检索：把每个物品与用户向量算一遍点积，排序取前 K。等这一步正确，再换成近似最近邻（ANN），用可能漏掉部分精确近邻的代价换速度。

要分清两种 Recall：**ANN 检索召回率**检查“找回多少精确近邻”，**推荐 Recall@K**检查“命中多少测试相关物品”。前者高，不代表向量模型一定懂用户兴趣。

8.3 在向量坐标里找更匹配的电影



点积相同，不代表方向或余弦相同。

点积比较

$u \cdot A = 1$; $u \cdot B = 1$

余弦相似度比较

$\cos(u, A) = 1$

$\cos(u, B) = 1 / \sqrt{2} = 0.707$

同样的点积，不同的余弦。u 与 A 都是 [1,0]，B 是 [1,1]：两项点积都是 1，余弦却分别为 1 与约 0.707。坐标只是教学例子，不能直接当作点击概率。

用户向量 $e_u = [1, 0]$，物品 A 为 $[0.8, 0.6]$，物品 B 为 $[0, 1]$。点积分别为 0.8 和 0，因此 A 得分更高。若使用余弦相似度，A 为 0.8，B 为 0，结论相同，但其他样本中两种度量可能产生不同排序。

8.4 常见误区与注意事项

- **随机负样本太容易。** 模型可能只学会区分热门与冷门，而无法区分细粒度兴趣。
- **难负样本可能是假负例。** 用户未点击不代表真正不相关。
- **批次内负样本受热度影响。** 热门物品频繁作为负例，可能产生不合理压制，需要采样校正或专门处理。
- **点积与余弦不可混用。** 点积受向量模长影响，余弦更强调方向。
- **双塔交叉能力有限。** 用户塔和物品塔通常在最后才交互，不适合所有精细排序任务。

本章小结

- 双塔模型分别编码用户和物品，输出同维向量；物品向量可离线计算，线上通过 ANN 检索。
- 点积受向量长度影响，余弦只关注方向；训练与检索必须使用同一种相似度定义。
- 负样本决定模型学到什么：随机负样本太容易，批次内负样本受热度影响，难负样本可能包含假负例。
- 双塔的交叉能力有限，更适合召回而不是精细排序；召回评估还要关注覆盖率和长尾。

关键术语：Embedding、双塔、点积、余弦相似度、负采样、ANN、假负例。

8.5 习题

1. 为什么双塔模型适合大规模召回，而复杂交叉模型更常用于较小候选集排序？
2. 使用 in-batch negatives 时，热门物品可能受到什么影响？
3. 用户向量为 $[1, 0]$，两个物品向量分别为 $[1, 0]$ 与 $[1, 1]$。点积相同，余弦相似度也相同吗？

▼ 参考答案

1. 双塔可预先计算物品向量并用 ANN 检索，线上无需对全库逐一运行复杂交叉；复杂交叉通常需要逐候选计算。
2. 热门物品更频繁出现在批次中并被当作负例，可能被过度压低，需要结合采样分布进行校正。
3. 不同。两个点积都是 1，但余弦分别为 1 和 $1/\sqrt{2} \approx 0.707$；第二个物品向量更长，方向也不同。

第 9 章　深度排序模型：Wide & Deep、DeepFM 与注意力

本章学习目标

1. 能说清 Wide 部分与 Deep 部分分别擅长记忆还是泛化。
2. 能解释 DeepFM 如何同时建模低阶和高阶特征交互。
3. 能读懂候选感知注意力公式，并说明它为什么比固定平均更灵活。

先修知识：第 7 章 CTR 与特征交叉；基础神经网络，包括 Embedding、MLP 和 Softmax。

本章导读：本章模型较多，但共同目标只有一个：更有效地组合稀疏特征和历史行为。不要只背结构图，要能说出每个模块解决了什么限制。

9.1 同样的特征，能不能组合得更好

深度排序模型将稀疏 ID、数值统计、序列和上下文特征联合起来学习：

- **Wide & Deep**：Wide 部分记忆显式规则与交叉，Deep 部分学习泛化表示。
- **DeepFM**：用 FM 建模低阶交互，用深度网络建模高阶交互，并共享特征 Embedding。
- **DIN 类注意力模型**：根据当前候选物品，对用户历史行为分配不同权重，形成与候选相关的兴趣表示。

9.2 这次推跑鞋，该参考哪段历史

先想象小林逛过“跑鞋、炒锅、手机、运动手表”。现在要推荐跑鞋，你可能会更重视运动相关的历史；改推炒锅时，又会重新分配注意力。候选感知注意力就在学这种“这次该参考哪段历史”。

下面先用容易理解的 **Softmax 归一化注意力**演示；它是通用形式，不是 DIN 原论文的精确公式。对历史行为 $h_1, \dots, h_T$：

$$v_u(i) = \sum_{t=1}^{T} \alpha_t(i) h_t, \qquad \alpha_t(i) = \frac{\exp a(h_t, i)}{\sum_s \exp a(h_s, i)}$$

公式参数说明

符号	含义	取值或维度
i	当前待排序的候选物品	物品编号或物品向量
t	用户历史行为的时间步	$1, \dots, T$
T	参与计算的历史行为数量	正整数；可截断以控制成本
h_t	第 t 个历史行为的表示向量	d 维列向量
$v_u(i)$	在候选为 i 时，用户 u 的动态兴趣向量	d 维列向量
$\alpha_t(i)$	候选为 i 时，第 t 个历史行为的注意力权重	$[0, 1]$，且对所有 t 求和为 1
$a(h_t, i)$	注意力打分函数，衡量历史行为与候选物品的相关性	实数
$\exp(\cdot)$	指数函数，保证权重为正	正实数
s	求和时遍历历史行为的临时索引	$1, \dots, T$
$\sum_s \exp a(h_s, i)$	所有历史行为打分的指数和，用于归一化	正实数

公式在说什么

先用打分函数 a 计算每个历史行为与当前候选物品的相关程度，再通过 Softmax 把这些分数变成总和为 1 的权重。最后对所有历史向量做加权平均，得到“针对这个候选物品”的用户兴趣向量。

若所有 $\alpha_t(i)$ 都等于 $1/T$，公式退化为简单平均；注意力机制的价值在于允许不同候选使用不同权重。例如候选是跑鞋时，跑鞋和运动手表的历史行为权重可以更高；候选是炒锅时，这些行为权重会下降。

与 DIN 原论文的区别：DIN 的局部激活单元采用 $v_u(i) = \sum_t a(h_t, e_i) h_t$，不强制权重之和为 1，以保留兴趣强度；其中 e_i 是候选物品向量。不要把这里用于入门的 Softmax 版本当作所有 DIN 实现的要求。参见 DIN 原论文第 4.3 节。

注意力权重也不是因果解释。实际影响还取决于历史向量的大小、方向与后续网络；权重大不能单独证明“贡献最大”，更不能证明它导致了点击。

9.3 同一个用户，可以有不同的兴趣重点

用户过去浏览过“跑鞋、炒锅、手机、运动手表”：

- 候选为跑鞋时，模型应更关注跑鞋和运动手表。
- 候选为炒锅时，跑鞋行为的相关性下降。

固定平均所有历史行为可能稀释当前兴趣，注意力机制可缓解这一问题。

9.4 常见误区与注意事项

- **复杂模型不一定优于良好基线。** 数据质量、特征、采样和评估可能更关键。
- **注意力权重不等于解释结论。** 应结合向量、后续计算与消融分析，不能凭一个权重断言影响大小或因果。
- **共享 Embedding 需保持一致。** DeepFM 等结构若实现时意外复制参数，可能改变原设计。

- **序列长度和延迟需权衡。** 更长历史不必然更好，也会增加噪声与计算成本。
- **深度模型同样需要正则化。** 包括早停、Dropout、权重衰减和低频特征处理等。

本章小结

- Wide & Deep 联合训练宽线性部分和深度网络：前者偏记忆显式交叉，后者偏泛化到未见组合。
- DeepFM 共享特征 Embedding，用 FM 建模低阶交互、用深度网络建模高阶交互。
- 候选感知注意力根据当前候选动态加权用户历史，使同一个用户面对不同候选时拥有不同兴趣向量。
- 模型越复杂，越需要可靠基线、正则化、负采样和线上约束；注意力权重不能直接当作因果解释。

关键术语：Wide & Deep、DeepFM、DIN、注意力、记忆、泛化、Embedding 共享。

9.5 习题

1. Wide & Deep 中两部分的职责有何差异？
2. 小林既买过跑鞋，也买过炒锅。今天分别给他推荐运动手表和汤锅，为什么不宜始终对所有历史取同一个简单平均？
3. 深度模型离线提升很小时，应优先检查哪些方面？

> ▼ **参考答案**
>
> 1. Wide 更偏向记忆显式特征及交叉，Deep 更偏向学习稠密表示和泛化交互。
> 2. 它可随候选变化重新分配历史行为权重，而简单平均对所有候选使用同一兴趣向量。
> 3. 应检查数据泄漏、样本和标签、特征质量、负采样、基线实现、评估划分、超参数及线上约束，而非只继续扩大网络。

第 10 章　序列推荐与用户兴趣演化

> **本章学习目标**
>
> 1. 能把下一物品预测表述为“给定历史序列，给候选物品打分”。
> 2. 能解释隐藏状态、物品表示和序列编码器之间的关系。
> 3. 能识别序列任务中的未来信息泄漏，并说明短期兴趣与长期兴趣的区别。
>
> **先修知识**：第 8 章 Embedding；第 9 章深度模型；了解 RNN 或 Transformer 的基本结构。
>
> **本章导读**：序列推荐增加的不是“更多特征”，而是“时间顺序”。因此数据切分、样本构造和因果掩码比模型结构更容易出错。

10.1 买完相机，再买什么

序列推荐根据有序行为预测下一次或下一批交互，适用于会话推荐、短视频、电商和音乐等场景。

典型任务为：给定 $i_1, i_2, \ldots, i_t$，预测 i_{t+1}。模型可包括 GRU、Transformer 及其他注意力结构。

10.2 把一串历史变成当前兴趣

模型将离散物品序列编码为隐藏状态，再对候选物品打分：

$$h_t = F(i_1, \ldots, i_t), \qquad s_{t+1,i} = h_t^T e_i$$

公式参数说明

符号	含义	取值或维度
$i_1, \ldots, i_t$	用户按时间排序的前 t 个物品交互	物品编号序列
t	当前时间步	正整数
F	序列编码器，例如 GRU 或 Transformer	参数通过训练得到
h_t	编码前 t 个行为后得到的隐藏状态	d 维列向量
i	候选物品编号	待预测的物品
e_i	候选物品 i 的表示向量	与 h_t 同为 d 维
$s_{t+1,i}$	给定历史后，候选物品 i 作为下一步交互的匹配分数	实数
$h_t^T e_i$	隐藏状态与物品向量的内积	实数

公式在说什么

F 把变长历史压缩成一个定长向量 h_t，再用内积给每个候选物品打分。分数越高，表示该物品越符合模型从历史中提取的当前兴趣。若要输出“给定候选集合下的下一物品概率”，可对这些候选分数做 Softmax。用采样候选训练还需考虑采样分布；对比学习本身不保证输出是校准概率。单独的 $s_{t+1,i}$ 不是概率。

注意：序列中的 i_t 是物品编号，e_i 是物品表示，两者不是同一个对象。预测 $t+1$ 时可以使用截至 t（包括 i_t）的历史，不能把 i_{t+1} 或之后的行为放进输入。

训练时可采用下一物品分类、对比学习或成对排序。像 SASRec 这样在一段序列的多个位置分别预测下一物品时，需要位置或时间信息，并用因果掩码阻止位置 t 看到后面的行为。例如输入 `[相机，镜头，三脚架]`，在“相机”位置预测“镜头”时，不能提前看到镜头。若只对一个已截断的历史前缀预测下一个物品，前缀内部的双向编码并不自动构成泄漏；关键是输入和训练目标是否匹配。

会话与长期历史不同：会话更强调当前意图；长期历史更强调稳定偏好。实际系统通常同时保留二者。

10.3 长期爱好和眼前任务，可能不一样

用户连续查看“相机 → 镜头 → 三脚架”，下一行为可能仍是摄影配件。若仅根据长期类别均值推荐，可能忽略当前购物任务；序列模型可更重视近期连续行为。

10.4 常见误区与注意事项

- **打乱顺序会破坏任务。** 序列模型的核心信息之一就是时间顺序。

- **预测下一物品时不能使用未来行为。** 训练样本构造和注意力掩码必须防止未来泄漏。
- **重复行为不一定都应保留。** 连续重复曝光、自动播放和主动点击的含义不同。
- **时间间隔很重要。** 一年前的点击与一分钟前的点击通常不应等价。
- **离线下一跳准确不等于长期推荐更好。** 还需观察列表质量和长期指标。

本章小结

- 序列推荐利用行为顺序，把历史 $i_1, \ldots, i_t$ 编码为隐藏状态 h_t，再给候选物品打分。
- 下一物品预测不能看见目标与未来行为；SASRec 式逐位置训练需用因果掩码，具体编码约束取决于样本和目标。
- 短期会话反映当前意图，长期历史反映稳定偏好，实际系统通常同时建模。
- 下一跳准确率不等于长期推荐质量；序列质量、时间间隔和重复曝光都需要处理。

关键术语：序列推荐、会话、隐藏状态、下一物品预测、因果掩码、短期兴趣、长期兴趣。

10.5 习题

1. 序列推荐与普通基于用户统计特征的推荐相比，新增了什么信息？
2. 用输入 `[相机, 镜头, 三脚架]` 做逐位置下一物品训练时，在“相机”位置预测“镜头”，为什么需要遮住后面的输入？
3. 为什么短期会话兴趣和长期兴趣通常需要同时建模？

> **▼ 参考答案**
>
> 1. 行为顺序、相对位置及可进一步加入的时间间隔等动态信息。
> 2. 因为输入后面已经放着答案“镜头”，不遮住就可能直接看到答案；因果掩码让该位置只读取当时已知的前缀。
> 3. 短期兴趣反映当前任务或意图，长期兴趣反映稳定偏好；只使用任一方都可能降低推荐质量。

第 11 章　学习排序：Pointwise、Pairwise 与 Listwise

> **本章学习目标**
>
> 1. 能区分 Pointwise、Pairwise 和 Listwise 的训练单位与优化目标。
> 2. 能读懂 Pairwise 对数损失，并解释分数差的正负如何影响损失。
> 3. 能说明 NDCG 为什么不能像交叉熵一样直接求导，以及排序评估为何要按用户分组。
>
> **先修知识**：第 4 章 NDCG 与 AUC；第 7 章打分模型；Sigmoid 与交叉熵。
>
> **本章导读**：排序学习关注的不是“每个物品各自准不准”，而是“同一组物品之间的相对顺序对不对”。

11.1 是逐个打分，还是直接比较先后

假设三部电影都能打分，接下来你更想教模型哪件事："A 应该打几分"，还是"A 应排在 B 前面"，或者"整张片单应该怎样排"？这对应学习排序（Learning to Rank, LTR）的三个常见视角：

- **Pointwise**：将每个用户—物品对独立转换为分类或回归问题。
- **Pairwise**：学习"物品 A 应排在物品 B 前"的相对关系。
- **Listwise**：以整个列表为训练单位，直接优化某种列表级目标或其近似。

范式	一条训练样本	标签形式	典型损失	主要优点	主要局限
Pointwise	一个用户—物品对 (u, i)	点击 0/1、评分、是否转化	交叉熵、均方误差	实现简单，可复用普通分类器	不直接约束物品间顺序
Pairwise	同一用户下的物品对 (u, i, j)	$i \succ j$ 的偏好关系	成对逻辑损失、Hinge	直接学习相对顺序	物品对数量大，需要采样
Listwise	一个用户/请求对应的候选列表	排列、相关性等级	列表概率交叉熵、NDCG 代理损失等	与列表指标更接近	训练和实现更复杂；使用 Softmax 本身不足以判定为 Listwise

11.2 该排前面的，却排到了后面

Pairwise 常优化：

$$L = -\log \sigma(s_i - s_j), \qquad i \succ j$$

公式参数说明

符号	含义	取值或说明
L	一个物品对上的成对排序损失	非负实数，越小越好
i	应排在前面的物品，通常是相关或正反馈物品	物品编号
j	应排在后面的物品，通常是负样本	物品编号
s_i, s_j	模型给物品 i, j 的排序分数	同一模型下的可比实数
$s_i - s_j$	两个物品的分数差	实数
$i \succ j$	排序约束，表示 i 应优先于 j	关系符号，不是数值大小
$\sigma(\cdot)$	Sigmoid 函数	输出 $(0, 1)$
$\log$	自然对数	底数为 e

公式在说什么

若正物品 i 的分数高于负物品 j，分数差为正，$\sigma > 0.5$；只有分数差足够大时，σ 才接近 1，损失才接近 0。若顺序反了，分数差为负，损失变大。因此该损失直接惩罚"把该排在前面的物品排到后面"。

数值直觉

若 $s_i - s_j = 0.2$：

$$-\log \sigma(0.2) \approx 0.598$$

若顺序反转，$s_i - s_j = -0.2$：

$$-\log \sigma(-0.2) \approx 0.798$$

差值越大，两种排序的损失差距越明显。Listwise 方法可能直接考虑整个排列的概率或与 NDCG 相关的目标。

NDCG 含排序操作，本身不可直接常规求导，因此许多方法使用可微近似、代理损失或其他优化技巧。

11.3 用两个候选看懂成对损失

若相关物品 A 的分数为 0.9，不相关物品 B 为 0.7，则 $s_A - s_B = 0.2$，成对损失约 0.598；反过来则约 0.798。这里只是后一种惩罚更大，不表示前一种已经足够好。Pointwise 则可能分别拟合 A 的标签 1 和 B 的标签 0。

11.4 常见误区与注意事项

- **分类最优不等于排序最优。** 应明确训练损失是否贴近业务指标。
- **Pairwise 对数量可能很大。** 需要采样，并控制训练成本。
- **Listwise 不一定总是更好。** 数据、标签粒度和优化难度都会影响结果。
- **不同查询或用户的列表不宜随意混合。** 相对比较应在合理分组内进行。
- **头部指标需重点关注。** 全局排序改善可能并未改善用户实际看到的前几项。

本章小结

- Pointwise 独立预测每个物品，Pairwise 学习两个物品的相对顺序，Listwise 直接优化整个列表。
- Pairwise 对数损失随“正分减负分”增大而减小；正物品仅高一点时，损失还不会接近 0。
- NDCG 含不可微的排序操作，通常用代理损失或近似方法优化。
- 排序比较必须按用户或查询分组，不能把不同用户的物品混在同一列表中比较。

关键术语：Learning to Rank、Pointwise、Pairwise、Listwise、成对损失、分组排序。

11.5 习题

1. CTR 逻辑回归通常属于哪一类 LTR 思路？
2. Pairwise 训练为什么通常需要构造或采样物品对？
3. 为什么不能简单把 NDCG 当作普通交叉熵一样直接优化？

▼ **参考答案**

1. 通常属于 Pointwise，因为每个曝光样本独立预测点击概率。
2. 全量物品对数量增长很快，且许多对信息量低，因此常采样以提高效率和聚焦有效比较。
3. NDCG 依赖离散排序和位置，排序操作不可微，通常需代理目标或专门优化方法。

第 12 章　多目标、多任务学习与概率校准

> **本章学习目标**
>
> 1. 能说明为什么真实推荐不能只优化点击率。
> 2. 能读懂多任务加权损失，理解共享参数与任务冲突带来的负迁移。
> 3. 能区分排序能力与概率校准，并说明多目标融合前为什么要统一尺度。
>
> **先修知识**：第 4 章 AUC 与 LogLoss；第 7 章 CTR 预估；第 9 章深度模型。
>
> **本章导读**：本章的核心不是某个网络结构，而是“业务目标如何转化为可优化、可评估的数学目标”。

12.1 点得多、看得久、买得多，怎么取舍

真实推荐往往同时关注点击、转化、观看时长、互动、退货和长期满意度。常见做法包括：

- 分别训练多个模型，再通过公式融合分数。
- 使用多任务模型共享表示，同时输出多个目标。
- 在重排阶段引入多样性和业务约束。

概率校准关注预测概率是否与真实发生率一致。例如所有预测为 0.2 的样本，实际点击比例是否接近 20%。

12.2 多门功课，怎样一起训练

多任务训练常写为：

$$L = \sum_{m} \lambda_m L_m$$

公式参数说明

符号	含义	取值或说明
L	多任务模型的总训练损失	实数，越小越好
m	任务编号，例如点击、购买、有效观看	离散索引
L_m	第 m 个任务自己的损失	非负实数
λ_m	第 m 个任务的损失权重	非负实数；越大表示越重视该任务

公式在说什么

总损失是各任务损失的加权和。训练时，梯度会同时来自多个任务。若某个 λ_m 很大，模型会更努力优化该任务，但也可能牺牲其他任务。权重不是模型自动学到的业务价值，需要结合验证集和线上实验确定。

共享层可以学习通用知识，但目标冲突时可能产生负迁移，因此可使用独立输出层、专家网络或更复杂的任务关系建模。

分数融合如：

$$S = w_1\hat{p}_{click} + w_2\hat{p}_{buy} + w_3\hat{v}_{quality}$$

公式参数说明

符号	含义	取值或说明
S	用于最终排序的综合分数	实数，通常越大越优先
$\hat{p}_{click}$	模型预测的点击概率	$[0,1]$，但需校准
$\hat{p}_{buy}$	模型预测的购买概率	$[0,1]$，但需校准
$\hat{v}_{quality}$	内容质量、观看时长或其他价值指标的归一化分数	通常先归一化到固定范围
w_1, w_2, w_3	三个目标的融合权重	由业务和实验确定，通常非负

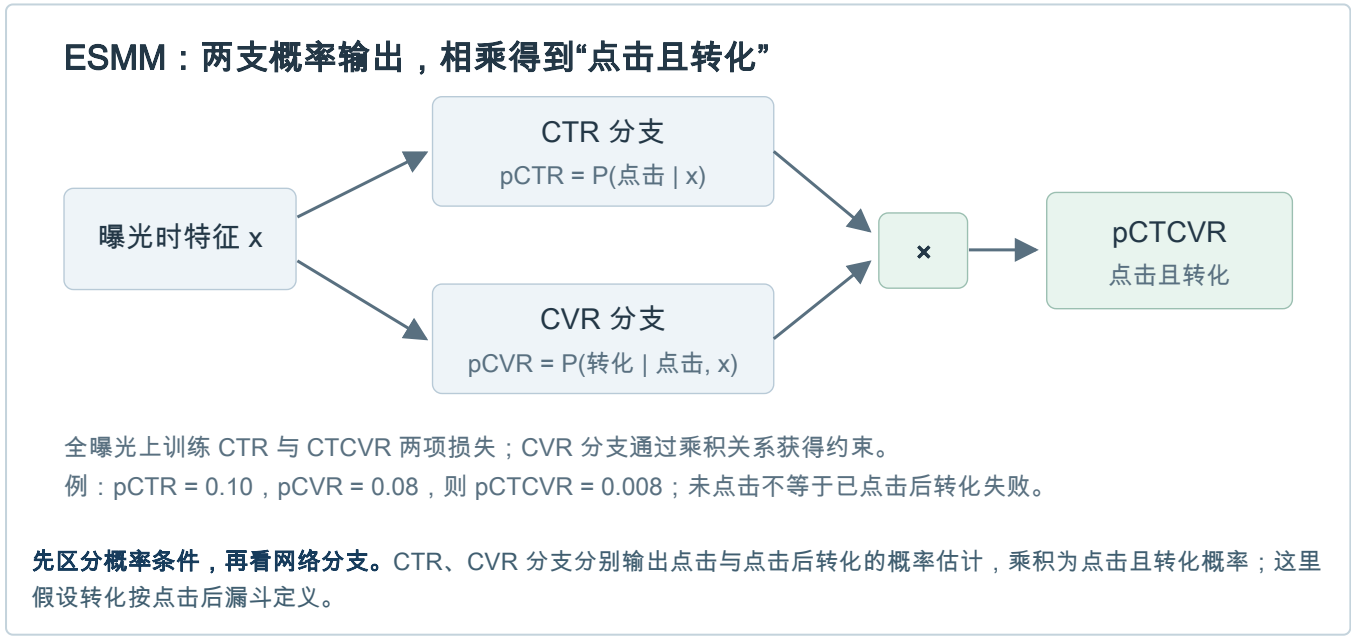
公式在说什么

融合公式像把不同功课折算成总成绩：每一项先明确怎么计分，权重再表达你重视多少。若 CTR 约为 0.1，观看时长约为 600 秒，等权直接相加，时长很容易在数值上占主导。

可以对时长做变换、对概率检查校准，也可以用带单位含义的权重折算价值。归一化是在改尺度，校准是在检查概率是否对应发生率，二者不是同一个动作，也不要求所有目标有相同分布。

必须注意各分数的量纲、分布和校准状态，不能直接假设它们可比。

12.3 点开了，却很快退出：到底算不算成功



先区分概率条件，再看网络分支。CTR、CVR 分支分别输出点击与点击后转化的概率估计，乘积为点击且转化概率；这里假设转化按点击后漏斗定义。

一条视频封面吸引人，点击率很高，用户却看两秒就离开。只奖励点击，模型可能越来越擅长挑这种视频。加入有效观看或满意度目标，相当于告诉它：“请把人吸引进来，也让人愿意留下。”不过两个目标可能冲突，需要通过实验确定取舍。

做广告时，还要问一句“转化率的分母是谁”：

- CTR：曝光后点击的概率。
- CVR：通常指**已点击条件下**转化的概率，具体口径需写清。
- CTCVR：曝光后既点击又转化的概率。

在转化按“先点击再转化”定义、且条件特征一致时，有 $P(\text{点击且转化} \mid x) = P(\text{点击} \mid x)P(\text{转化} \mid \text{点击}, x)$。所以 $\hat{p}_{buy}$ 到底是 CVR 还是曝光后的购买概率，不能省略。ESMM 等模型需要点击与转化标签，只有一个点击标签的数据不能直接照搬。

12.4 常见误区与注意事项

- **权重不是纯技术参数。** 它体现业务价值取舍，应结合实验确定。
- **不同目标可能冲突。** 点击、时长和转化并不总是同向变化。
- **排序正确不代表概率校准。** 一个模型可保持相同排序，却整体高估 CTR。
- **采样可能改变概率尺度。** 校准集需要尽量代表目标曝光分布，并与最终测试集分开；只在同样有偏的采样集上校准，不能恢复真实线上 CTR。若真实分布和采样比例都未知，就如实报告这一限制。
- **多任务并非必然优于单任务。** 数据不足或目标冲突时可能负迁移。

本章小结

- 推荐通常同时优化点击、转化、时长、留存等多个目标，这些目标可能相互冲突。
- 多任务损失是各任务损失的加权和；共享表示能提升效率，但可能产生负迁移。
- AUC 衡量排序，概率校准衡量预测概率是否接近真实发生率，两者不能互相替代。
- 多目标融合要明确量纲、概率的条件事件和权重含义；可通过校准、变换或有单位的权重处理尺度，不要求各目标分布完全相同。

关键术语：多任务学习、多目标优化、损失权重、负迁移、概率校准、分数融合。

12.5 习题

1. 为什么不能未经处理就把 CTR、转化率和观看时长相加？
2. 多任务模型出现某任务提升、另一任务下降，可能是什么原因？
3. 概率校准与 AUC 是否是同一回事？

> **▼ 参考答案**
>
> 1. 三者量纲、取值范围和分布不同，直接相加会使某些目标因数值尺度而主导结果。
> 2. 可能任务目标冲突、共享表示不适合所有任务、损失权重不合理或数据质量不同。
> 3. 不是。AUC 关注正负样本相对排序；校准关注预测概率数值是否对应真实发生率。

第 13 章　冷启动、探索利用与强化学习入门

> **本章学习目标**
>
> 1. 能区分新用户、新物品和新系统三类冷启动，并提出对应的初始策略。
> 2. 能解释探索与利用的冲突，以及 ϵ-greedy、UCB、Thompson Sampling 的基本思想。
> 3. 能说明多臂老虎机与完整强化学习的区别，知道离线评估新策略的困难。
>
> **先修知识**：第 5～6 章协同过滤与隐式反馈；基础概率分布。
>
> **本章导读**：冷启动不仅是“没有数据”，更是“旧策略决定未来能看到什么数据”的反馈回路问题。

13.1 新用户来了，推荐器还不认识他

冷启动包括：

- 新用户：缺少历史行为。
- 新物品：缺少交互与协同信号。
- 新系统：整体数据不足。

探索与利用是推荐中的持续权衡：利用当前最优估计可获得短期收益，探索不确定内容可获得信息并可能改善长期效果。二者的区别可以用一个简单问题理解：你已经知道一家餐厅不错，是继续去这家（利用），还是尝试一家评价不多但可能更好的新店（探索）？

策略	核心做法	优点	主要风险
ϵ-greedy	以 ϵ 的概率随机探索，其余时间选择当前最优	简单、易实现	探索完全随机，可能浪费流量
UCB	选择“平均收益 + 不确定性奖励”最高的选项	有方向地探索高不确定选项	奖励尺度与探索系数难调
Thompson Sampling	从收益的后验分布中采样，按采样值选择	不确定性建模自然	需要合适的概率模型与计算

在最简单的多臂老虎机例子里，“一个臂”可以理解成一家可选餐厅：选一家吃饭，得到一次反馈，再更新估计。这三种策略通常只决定当前选谁，不显式建模用户状态如何随推荐改变；完整强化学习还会考虑状态转移和长期回报。

例如 $\epsilon = 0.1$ 时，每次以 10% 概率探索，90% 概率选当前估计最好的选项；100 次决策中探索次数的期望是 10 次，并非每次运行都恰好 10 次。探索还可能随机选中当前最优项，因此“探索次数”不等于“选了非最优项的次数”。

13.2 去熟悉的店，还是试试新店

冷启动常用方法：

- 新用户：人口属性、上下文、兴趣选择、热门与高质量内容。

- 新物品：文本、图像、类别、作者等内容特征。
- 混合策略：内容模型提供初始表示，行为积累后逐渐引入协同信号。

上一节已经比较三种多臂老虎机策略的核心差异。实际选择时，还要考虑奖励尺度、流量成本、反馈延迟和可解释性：探索预算太少可能发现不了新内容，探索预算太多则会损害当前用户体验。

强化学习进一步考虑状态、动作、奖励和长期回报，但建模、离线评估和在线安全都更复杂。

13.3 新商品没有点击，也不该永远没机会

新上架商品没有点击数据，可根据标题、图片、类目和品牌生成内容 Embedding，与相似成熟商品一起参与召回；获得少量曝光后，再根据真实反馈调整，而不是始终零曝光或一开始大量曝光。

13.4 常见误区与注意事项

- **不探索会强化旧偏差。** 新物品永远没有数据，系统会误以为它们不受欢迎。
- **探索必须控制成本。** 应设置流量、频控和质量约束。
- **老虎机不等于完整强化学习。** 前者通常不显式建模长期状态转移。
- **离线日志评估新策略存在反事实困难。** 未展示内容的反馈无法直接知道。
- **新物品扶持不等于无视质量。** 应兼顾用户利益和生态目标。

本章小结

- 冷启动分为新用户、新物品和新系统；内容特征、规则和热门内容是常用的初始信号。
- 探索与利用存在冲突：只利用会强化旧偏差，只探索会损害短期体验。
- ϵ-greedy、UCB 和 Thompson Sampling 都是多臂老虎机策略，重点在于如何量化并控制不确定性。
- 多臂老虎机通常不建模完整的状态转移，不能简单等同于强化学习；离线评估新策略还存在反事实困难。

关键术语：冷启动、探索与利用、多臂老虎机、ϵ-greedy、UCB、Thompson Sampling、离策略评估。

13.5 习题

1. 为什么纯协同过滤容易产生“新物品没有曝光—没有数据—继续没有曝光”的循环？
2. ϵ-greedy 中 ϵ 过大和过小分别可能带来什么问题？
3. 为什么内容特征对新物品冷启动特别重要？

> **▼ 参考答案**
>
> 1. 模型依赖行为共现，新物品缺乏行为导致难以被召回，又无法通过曝光积累行为，形成反馈回路。
> 2. 过大可能过多展示不确定或低质量内容，损害短期体验；过小则探索不足，难以发现潜在优质内容。
> 3. 新物品即使缺少行为，也通常已有文本、图像、类目等属性，可用于初始匹配和表示。

第 14 章　重排、多样性、在线实验与反馈偏差

本章学习目标

1. 能解释单物品分数最高的列表为什么未必是最好的列表。
2. 能读懂相关性—多样性权衡公式，并说明 λ 的作用。
3. 能识别位置偏差、流行度偏差和选择偏差，设计带护栏指标的 A/B 测试。

先修知识：第 4 章评估；第 12 章多目标与分数融合。

本章导读：本章从“给单个物品打分”上升到“优化整个列表”，同时开始处理推荐系统与用户之间的双向影响。

14.1 前十名都是同一个系列，怎么办

精排分数最高的前 K 个物品不一定组成最佳列表。例如它们可能全部属于同一作者或类别。重排通过去重、打散、多样性、新鲜度和约束控制改善整体体验。

A/B 测试将用户或流量随机分配到不同策略，用于估计策略对真实指标的影响。常见护栏指标包括延迟、崩溃率、投诉率和核心体验指标。

14.2 给重复扣点分，再检验真实效果

MMR：已经选了 A，下一部选 B 还是 C？

本例 λ = 0.5；候选相关性与相似度都在 0 ~ 1。

候选	相关性 rel	与已选 A 的相似度	0.5 × rel − 0.5 × sim
B	0.90	0.95	−0.025
C	**0.85**	**0.10**	**0.375 ← 选择 C**

C 的相关性略低，但与已选电影重复更少，所以这一轮得分更高。
这是贪心选下一项的打分，不是概率；λ 接近 1 时更重视相关性。

用数字检查相关性与多样性的取舍。λ 的数值含义按本图的 MMR 公式理解，不要与正文中其他多样性目标的惩罚系数混用。

一种简化的多样性目标是：从候选中选出恰好 K 个不同物品，让所选集合 L 的效用尽量大：

$$S(L) = \sum_{i \in L} rel_i - \lambda \sum_{\substack{i,j \in L \\ i<j}} sim(i,j), \qquad |L| = K$$

公式参数说明

符号	含义	取值或说明
$S(L)$	列表 L 的整体重排效用	实数，越大越好
L	当前选择的 K 个物品	此简化公式只决定集合，不区分内部顺序
i, j	列表中的两个不同物品	$i < j$ 表示只计算一次无序物品对
rel_i	物品 i 与用户的相关性分数	通常已归一化
$sim(i, j)$	物品 i 与 j 的相似度，例如内容或类别相似度	通常为 $[0, 1]$
λ	多样性惩罚强度	非负实数；越大越强调多样性
$\sum_{i \in L} rel_i$	列表内所有物品相关性之和	非负实数
$\sum_{i<j} sim(i, j)$	列表内所有不重复物品对的相似度之和	非负实数

公式在说什么

第一项奖励相关性，第二项给重复内容“扣分”。固定选 K 个，$\lambda = 0$ 时会选相关性最高的 K 个物品；增大 λ，就更愿意牺牲一点相关性来换不同口味。注意：这个公式对集合内部换顺序不敏感，不能单独决定最终展示顺序。选完后可先按相关性排；更精细的方法再加入位置权重。

例如要选两部电影：A、B、C 的相关性分别为 0.90、0.88、0.85；A 与 B 同系列，相似度 0.95；A 与 C 不同类型，相似度 0.10。取 $\lambda = 0.2$，选 A+B 得 $1.78 - 0.19 = 1.59$，选 A+C 得 $1.75 - 0.02 = 1.73$。虽然后者相关性总和略低，却因重复少而更优。

这个子集目标也不适合穷举所有组合。可以用贪心方法：每次加入一个“新增相关性减新增相似性惩罚”最大的物品。MMR、DPP 是另外常见的多样性方法，它们各自有不同目标，不要视为这个公式的同义写法。

反馈数据常见偏差：

- **位置偏差**：靠前的物品更容易被点击。
- **选择偏差**：训练数据只覆盖旧策略选择展示的内容。
- **流行度偏差**：热门物品获得更多曝光，从而继续积累更多数据。
- **幸存者偏差**：只看到留下用户的行为，忽视流失用户。

14.3 点击涨了，投诉也涨了，能算成功吗

某实验使 CTR 提升 8%，但投诉率明显上升、次日留存下降。不能仅凭 CTR 宣布全量，应结合实验设计、统计显著性和长期或护栏指标判断。

14.4 常见误区与注意事项

- **实验随机化单位要一致。** 通常按用户分流可减少同一用户跨策略污染，但具体取决于业务。
- **不要只看整体平均。** 新老用户、不同活跃度群体可能反应相反。
- **注意新奇效应与周期效应。** 实验过短可能误判。
- **重排规则可能伤害相关性。** 多样性权重需要实验验证。
- **离线数据无法完整回答反事实问题。** “若当时展示另一物品会怎样”通常没有直接标签。

本章小结

- 精排优化单个物品的相关性，重排优化整个列表的多样性、去重、新鲜度和业务约束。
- 相关性—多样性目标通过 λ 权衡；λ 越大，越愿意牺牲相关性换取列表差异。
- 位置偏差、流行度偏差和选择偏差会污染训练数据，使模型把曝光优势误学成用户偏好。
- A/B 测试要明确随机化单位、实验周期和护栏指标，不能只看 CTR 的提升。

关键术语：重排、多样性、位置偏差、流行度偏差、选择偏差、A/B 测试、护栏指标。

14.5 习题

1. 前 10 个候选都是同一作者的相似视频。即使每条精排分数都很高，整张列表为什么仍可能需要调整？
2. 某策略 CTR 提升但留存下降，应如何理解？
3. 位置偏差如何影响训练数据？

> **▼ 参考答案**
>
> 1. 列表可能高度重复或违反约束；重排可优化整体多样性、新鲜度和业务规则。
> 2. 点击提升可能以内容质量或长期体验为代价，应结合指标定义、统计检验和长期结果综合判断，而非单指标决策。
> 3. 靠前物品获得更多曝光和点击，点击标签同时反映位置与兴趣，若直接训练会把位置优势误学为偏好。

第 15 章　综合实践：从 MovieLens 到完整推荐 pipeline

> **本章学习目标**
>
> 1. 能把前十四章的知识组织成一条可运行的推荐流水线。
> 2. 能使用统一的数据切分和指标，公平比较热门、ItemCF、矩阵分解和双塔等模型。
> 3. 能进行分层误差分析，识别热门偏置、冷启动和评估协议问题。
>
> **先修知识**：入门版读完第 1、3～5 章并会基础 Python 即可；进阶版再补对应模型章节。
>
> **本章导读**：本章不追求最复杂的模型，而追求“结果可复现、比较公平、错误可解释”。这是从学习者走向工程实践的关键一步。

15.1 先做一件小事：给老用户推荐没评过的电影

第一个项目只完成“读数据 → 分时间 → 热门推荐/ItemCF → 推荐 10 部 → 算指标”。先不做网页、接口、深度模型，也能形成完整的离线实验。

使用 MovieLens 100K，解压后先看 `u.data`。按官方 README，每行由制表符分隔，依次为：

```
user_id    item_id    rating    timestamp
```

实际文件没有表头。`rating` 是 1～5 分，`timestamp` 是 Unix 时间戳。它记录的是**评分**，不是“每次曝光后有没有点击”，所以这里做的是一个用高分当作喜欢信号的 Top-K 练习，不称为真实 CTR 实验。

15.2 入门版：把实验规则写具体

下面是一套教学协议，不是唯一标准。先固定它，再比较算法：

1. **先按全局时间切分所有评分。** 根据 `timestamp` 的 70%、85% 分位点取两个边界：训练为 $t < t_1$，验证为 $t_1 \le t < t_2$，测试为 $t \ge t_2$。同一时间戳不拆到两边，实际比例可略有偏差。后续只用验证集选参数。
2. **再定义“喜欢”。** 各分区里评分 ≥ 4 记作正交互；低分保留为评分记录，但不塞进这个简化 ItemCF 的正交互集合。空白表示没观察到，不补成 0 分或明确负反馈。
3. **只用训练期建模型。** 热门榜按训练期正交互用户数统计；ItemCF 的共同用户、相似度也只看训练期。候选目录限定为训练期至少有一个正交互的电影，避免让算法偷偷使用未来交互。
4. **把已评过的电影过滤掉。** 每个用户过滤其训练期所有评分对应的电影，包括低分电影。这样不会把用户已经打过 1 分的电影又推荐回来。
5. **确定评估对象。** 先评训练期至少有一个正交互的用户。相关集合为该用户验证/测试期的高分电影，与候选目录取交集，再减去训练期已评集合。交集后为空的用户不算 Recall/NDCG，但记录人数和原因；未来首次出现的用户、训练目录外的新电影单独统计，留给冷启动实验。
6. **给所有剩余候选打分。** 暂时不采样评估负例。热门榜直接按热度排；ItemCF 用第 5.2 节的相似度求和。无相似信号时用训练期热度兜底，平局统一按物品 ID 排。
7. **每人取前 10，再算平均。** 用第 4 章的二值相关性计算 Recall@10、NDCG@10，然后对有效用户取宏平均。候选不足 10 时不重复填充；Recall 用真实相关集合个数作分母，NDCG 的 IDCG 按同一评估范围计算。
8. **最后测试一次。** 本入门版选定参数后仍保持训练期模型和候选目录不变，不把验证交互加进训练。验证集与测试集都以训练期历史作为输入；记录两者间隔造成的历史陈旧限制。这便于先学会控制变量，之后再扩展“训练+验证重训”或逐时更新，并重新说明协议。

这套规则有意先聚焦“训练期已知用户、已知电影”的暖启动场景。报告里必须带上有效用户数、新用户数、新电影数和被排除的相关交互数，否则一个漂亮的平均分可能只是因为把难题删掉了。

做到什么算完成？ 两个基线使用相同的候选与用户集合；你能查看某个用户的历史、前 10 部推荐、命中的电影，并解释 ItemCF 中哪部历史电影给候选贡献了分数。指标没提升也可以是有效结果，不需要为了好看反复改测试规则。

15.3 进阶版：每次只多加一块积木

完成入门版后，按需要选一步，不必全做：

下一步	想回答的问题	需要补什么
BPR-MF	学出的向量能否超过共现相似度？	第 6 章；固定训练负采样策略，保持评估不变
简化双塔	加入用户和物品特征能否改善召回？	第 8 章；先精确点积检索，再研究 ANN
LR / DeepFM 排序	在固定候选里，精细模型能否排得更好？	第 7、9 章；MovieLens 的合成负例只能定义为偏好分类实验
用真实曝光数据做 CTR	概率能否对应真实点击发生率？	曝光、点击、时间字段；重新定义标签、窗口与 AUC/LogLoss 协议
加内容兜底或类别打散	新电影怎么办？列表是否太重复？	第 13、14 章；同时记录覆盖与相关性变化

如果加入二阶段排序，候选应由训练期模型生成。**只对已召回结果算排序指标**和**对原始相关集合算端到端指标**要分开报告；召回漏掉的电影不能在端到端评估时从分母里悄悄删掉。

最小实验记录表（下面都留空，等待你的真实结果）：

实验	模型	参数/规则	有效用户数	Recall@10	NDCG@10	备注
A	热门推荐	正交互用户数	待填	待填	待填	基线
B	ItemCF	全邻居或邻居数	待填	待填	待填	同一评估范围
C（选做）	BPR-MF	维度、正则、采样、种子	待填	待填	待填	与 A/B 比较

另附：数据来源与校验值、时间边界、过滤前后规模、候选范围、平局与兜底规则、随机种子、Python/依赖版本、运行命令和耗时。固定种子有助复现，但不保证所有硬件上的浮点结果完全一致。

15.4 常见误区与注意事项

- 不要在测试集上反复选参数后再把同一结果作为最终成绩。
- 不要只报告总体均值，应检查不同用户和物品群体。
- 不要忽略已见物品过滤规则；规则不同，指标不可直接比较。
- 不要先追求复杂模型，再补数据和评估；顺序应相反。
- 不要把 MovieLens 的显式评分实验结论直接等同于工业隐式反馈场景。

本章小结

- 一个可复现的推荐项目应按“任务定义—时间切分—候选构造—基线—模型—统一评估—误差分析”推进。
- 热门推荐和 ItemCF 不是过时模型，而是判断复杂模型是否真正有效的必要基线。
- 实验记录必须包含数据版本、参数、随机种子和评估协议；协议不同的指标不能直接比较。
- MovieLens 适合学习显式反馈，但不能替代含曝光、点击和时间戳的工业隐式反馈数据。

关键术语：推荐流水线、基线、时间切分、候选集合、误差分析、可复现性。

15.5 综合习题

1. 设计一个实验，判断模型提升是否只来自“更偏向热门物品”。
2. 若矩阵分解训练损失下降，但 Recall@10 下降，列出至少三个检查方向。
3. 为什么不同论文或项目中的 Recall@10 不能在未检查协议时直接比较？

> **▼ 参考答案**
>
> 1. 固定数据和评估协议，对比模型与热门基线；统计推荐列表的平均热度、热度分布和覆盖率，并按冷热物品分层计算 Recall/NDCG。若指标提升仅伴随热门集中且长尾退化，说明提升可能主要来自流行度偏向。
> 2. 可检查：训练目标与 Recall 是否不一致、负采样是否不合理、是否过拟合、数据划分或评估是否泄漏、已见物品过滤是否错误、超参数是否不合适、验证与训练日志是否实现一致。
> 3. 数据版本、候选集合、相关标签、负样本数量、用户筛选、已见过滤和时间划分都可能不同，Recall@10 会随协议显著变化。

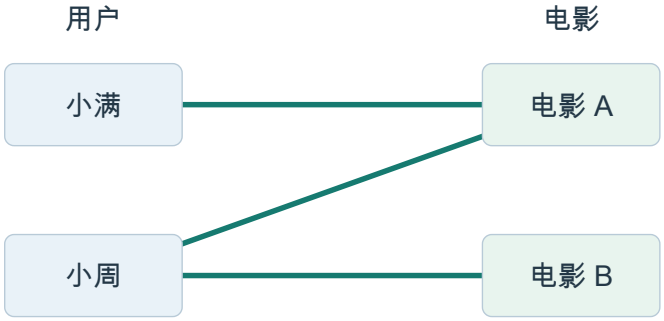
第 16 章　进阶专题：图推荐、混合推荐、多场景与生成式推荐

> **本章学习目标**
>
> 1. 能解释 LightGCN 的邻居聚合、层组合和过平滑问题。
> 2. 能区分混合推荐、上下文推荐、知识型推荐和多场景建模各自解决什么问题。
> 3. 能说明生成式推荐与传统级联推荐在建模对象、训练目标和工程约束上的差异，并判断其适用性。
>
> **先修知识**：第 5 ~ 14 章；图的基本概念、Softmax、Transformer 基础。
>
> **本章导读**：本章是选学内容，目标是建立判断力，而不是追逐最新模型名称。阅读时始终追问：它替代了原系统的哪一部分，代价是什么。

本章属于选学内容。建议在完成第 5 ~ 14 章并具备可复现实验后再阅读，用于衔接 fun-rec 的召回、重排与生成式推荐部分。前沿方法迭代较快，本章重点在于建立判断框架，而非追逐具体模型版本。

16.1 图推荐与 LightGCN

用户—物品二部图：沿三条训练期交互寻找新候选



小满 → A → 小周 → B

实线：已观察的训练交互

B 可成为小满的新候选

但共现不是“必然喜欢”

小满与 B 之间没有训练边；不能先把测试交互接上，再拿它评价传播效果。

只沿训练期已知关系传播。四个节点、三条实线构成一个最小二部图。图模型利用多跳关系，不代表已经观察到小满对电影 B 的反馈。

概念讲解

把用户和物品视为二部图中的两类节点，交互视为边，则推荐可以转化为图上的表示学习和链接预测问题。用户与物品的交互路径能够传播高阶协同信号：例如在路径 U—A—V—C 中，物品 C 与用户 U 没有直接交互，但其信息仍可通过中间节点影响 U 的表示。

关键原理

图卷积网络通过邻居聚合更新节点表示。以用户节点 u 为例，一种归一化聚合形式为：

$$e_u^{(k+1)} = \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(i)|}} e_i^{(k)}$$

公式参数说明

符号	含义	取值或维度
$e_u^{(k+1)}$	用户 u 在第 $k+1$ 层的表示向量	d 维列向量
k	当前图传播层数编号	从 0 开始
$e_i^{(k)}$	物品 i 在第 k 层的表示向量	d 维列向量
$\mathcal{N}(u)$	与用户 u 有交互的所有物品集合	集合
$\mathcal{N}(i)$	与物品 i 有交互的所有用户集合	集合
$\lvert\mathcal{N}(u)\rvert$	用户 u 的邻居数量	正整数
$\lvert\mathcal{N}(i)\rvert$	物品 i 的邻居数量	正整数
分母 $\sqrt{\lvert\mathcal{N}(u)\rvert\lvert\mathcal{N}(i)\rvert}$	归一化系数，降低高度节点的影响	正实数

公式在说什么

用户的新表示，等于其所有邻居物品上一层的表示加权求和。连接大量用户的物品会被除以更大的分母，避免热门物品主导所有用户的表示。$k = 0$ 时，$e_u^{(0)}$ 和 $e_i^{(0)}$ 是初始可学习 Embedding；每传播一层，就多融合一跳的协同信息。

物品节点采用对称更新：

$$e_i^{(k+1)} = \sum_{u \in \mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(i)||\mathcal{N}(u)|}} e_u^{(k)}$$

两式合起来，就是 LightGCN 的邻居聚合。它不使用特征变换矩阵，也不使用非线性激活，但仍然保留图结构和多跳传播。

LightGCN 的关键观察是：在协同过滤任务中，特征变换和非线性激活的贡献有限，因此仅保留邻居聚合，并将各层表示加权求和：

$$e_u = \sum_{k=0}^{K} \alpha_k e_u^{(k)}$$

公式参数说明

符号	含义	取值或说明
e_u	用户 u 的最终表示	d 维列向量
K	图传播的总层数	非负整数；K=0 只保留初始表示，与推荐列表的 K 无关
k	层编号	$0, \ldots, K$
α_k	第 k 层表示的组合权重	非负实数；常设为 $1/(K+1)$，也可学习
$e_u^{(k)}$	用户 u 在第 k 层的表示	d 维列向量

公式在说什么

最终向量不是只取最后一层，而是把第 0 层到第 K 层的结果加权平均。第 0 层保留初始可学习 ID 向量（不等同于文本、图片等原始内容特征），较深层吸收多跳协同信息。层组合有助于缓解只用最后一层带来的过平滑，但并不保证完全避免。

注意：层数 K 越大，传播范围越广，但过平滑和计算成本也越高。K 不能与评价指标中的 Recall@K 混为一谈。

这种简化减少了参数与训练难度，但并不意味着图模型在任何数据上都优于矩阵分解。图只能由当前训练截止时间前的交互构建；把测试边提前放进图传播，即使不拿它们算训练损失，也会泄漏。没有邻居的节点需要单独兜底。

示例

用户 U 交互过物品 A、B；用户 V 也交互过 A，并交互过物品 C。经过多层传播，C 的信息可以沿 U—A—V—C 路径影响 U 的表示，从而把 C 作为候选。该信号不是简单的共同邻居，而是多跳协同关系。

常见误区与注意事项

- **层数越深不一定越好。** 多层聚合可能导致过平滑，用户和物品表示趋于相似；常见有效层数较少，应通过验证集确定。
- **高度节点会主导聚合。** 热门物品连接大量用户，需要归一化、采样或热度校正。
- **图是动态的。** 新交互发生后图结构变化，离线图与在线图不一致会造成特征陈旧。
- **假负例仍然存在。** 图中未连接的物品不一定不相关，可能只是未曝光。
- **实现复杂度不等于效果。** 应先与 ItemCF、矩阵分解和双塔基线比较。

16.2 混合推荐、上下文感知与知识型推荐

概念讲解

单一方法各有盲区：协同过滤受冷启动影响，内容推荐可能重复，知识型推荐依赖规则和领域知识。混合推荐组合多种信号，以提高覆盖率和稳健性。上下文感知推荐进一步把时间、地点、设备、场景和用户当前任务纳入决策。知识型推荐则利用显式约束、规则或案例满足特定需求。

关键原理

常见混合方式包括：

- **加权混合**：对多个模型的分数加权融合。
- **切换混合**：根据数据条件选择不同推荐器，例如新用户使用内容推荐，老用户使用协同过滤。
- **级联混合**：前一模型产生候选，后一模型过滤或重排。
- **特征增强**：把内容、知识或上下文特征输入统一模型。
- **元级混合**：将一个推荐器学出的模型或表示交给另一个推荐器作为输入。另训模型来组合多个推荐器的预测分数，更常称为堆叠融合（stacking），不要混为同一定义。

知识型推荐可分为基于约束和基于案例两类。前者要求结果满足硬性条件，例如预算、库存、地区或兼容性；后者检索与当前需求相似的历史案例，再进行调整。

示例

用户首次访问租房平台，没有行为历史。系统先根据城市、预算、通勤时间和房型等硬约束过滤房源，再用内容特征与热门程度生成候选；用户产生浏览行为后，逐步提高协同过滤信号的权重。

常见误区与注意事项

- **混合不一定优于最强单模型。** 如果子模型高度相关，融合收益可能很小。
- **硬约束不能只作为排序特征。** 例如已下架或无库存的物品应直接过滤；预算若只是用户的软偏好，则可由排序权衡，不要把所有偏好都当硬约束。
- **上下文特征可能泄漏。** 使用“下单后填写的目的地”预测下单前行为，属于未来信息泄漏。
- **规则缺乏可扩展性。** 规则数量增加后，冲突处理、维护成本和可解释性都会恶化。
- **切换策略需要可观测条件。** 新老用户划分、行为数量阈值必须在线上稳定可得。

16.3 多场景与多域建模

概念讲解

同一产品往往包含多个场景，例如首页推荐、搜索推荐、信息流和购物车推荐。各场景的用户意图、候选集合和样本分布不同，但又共享用户长期偏好与物品知识。多场景建模的目标是同时利用共享信息和场景差异，而不是为每个场景完全独立建模型。

关键原理

典型结构包括共享底层、场景特定塔、专家网络和门控机制。以多门混合专家为例，第 k 个场景的门控权重为：

$$g^k(x) = \mathrm{softmax}(W_g^k x), \qquad o^k(x) = \sum_{j=1}^{M} g_j^k(x)\, E_j(x)$$

公式参数说明

符号	含义	取值或维度
k	场景编号，例如首页、搜索、信息流	离散索引
x	当前样本的输入特征	特征向量
W_g^k	第 k 个场景的门控网络参数	形状为 $M \times d_x$ 的矩阵
$g^k(x)$	第 k 个场景对各专家的权重向量	M 维，元素非负且和为 1
$g_j^k(x)$	第 k 个场景给第 j 个专家的权重	$[0, 1]$
j	专家编号	$1, \ldots, M$
M	专家网络总数	正整数
$E_j(x)$	第 j 个专家网络输出的中间表示	各专家输出需同维，不必与最终任务输出同维
$o^k(x)$	第 k 个场景的专家混合表示，之后可接场景塔和输出层	与专家输出同维的向量

公式在说什么

每个场景先用自己的门控网络给所有专家打分，再用 Softmax 得到权重；最终输出是各专家输出的加权组合。不同场景可以给同一个专家不同权重，因此既共享专家的通用知识，又允许场景差异。若门控对所有场景都一样，专家混合部分就不再区分场景；是否仍存在场景差异，还取决于后续是否保留独立的场景塔。

从简单方案开始：把场景 ID 作为输入，已经是一种多场景基线，并不一定要使用专家网络。若分场景评估显示负迁移或小场景效果差，再尝试独立场景塔、门控或采样加权。这里把每个场景对应一个门，是多任务专家结构的一种使用方式，并非所有多场景模型的定义。

示例

电商首页强调发现和多样性，搜索场景强调查询相关性。二者可以共享物品 Embedding 和用户长期兴趣，但为场景保留独立输出层或专家，避免搜索的强意图完全覆盖首页的浏览偏好。

常见误区与注意事项

- **先比较简单基线。** 共享模型加场景 ID、各场景独立模型与专家模型都值得对照；结构更复杂并不自动更合理。
- **场景样本量差异会造成训练倾斜。** 大场景可能主导共享参数，应进行采样、加权或分层评估。
- **负迁移确实存在。** 场景目标冲突时，强行共享可能损害某一场景。
- **必须分场景评估。** 总体指标提升可能由大场景贡献，小场景实际退化。
- **跨场景特征要注意时间一致性。** 其他场景的实时行为可用于当前请求时，必须确保线上同样可得。

16.4 生成式推荐

概念讲解

传统级联推荐以判别式打分为主：对候选物品分别估计点击率或偏好分数，再排序。“生成式推荐”覆盖不止一种做法。本节先讲 TIGER 一类容易形成直觉的路线：把物品变成若干离散 Token 组成的语义 ID，再根据用户历史逐步生成下一物品的 ID，像按层级缩小范围，最后落到一个具体商品。但不是所有生成式推荐都用语义 ID，语义 ID 的各个 Token 也不一定有可读的类目含义。

关键原理

设用户历史对应的物品 Token 序列为 $z_1, \ldots, z_t$，生成式模型学习：

$$P(z_{t+1} \mid z_1, \ldots, z_t)$$

公式参数说明

符号	含义	取值或说明
$z_1, \ldots, z_t$	用户历史被 Tokenizer 编码后的 Token 序列	每个 z 是离散 Token ID
t	当前已生成或已知的 Token 数量	正整数
z_{t+1}	下一个待预测 Token	词表中的离散 ID
$P(z_{t+1} \mid z_1, \ldots, z_t)$	在历史 Token 已知时，下一个 Token 的条件概率	$[0, 1]$
$\mid$	条件概率符号，表示“在……已知的条件下”	不是除法

公式在说什么

生成式推荐把“下一个买什么/看什么”转化为“下一个 Token 是什么”。模型一次预测一个 Token 的概率分布，生成完整语义 ID 后，再映射回真实物品。一个物品可能对应多个 Token，因此 Token 概率高不等同于物品推荐概率高。

训练可采用下一 Token 预测。推理时按概率生成 Token 序列，再映射回真实物品，并通过约束解码、合法 ID 校验和业务规则保证结果有效。部分方法使用大语言模型初始化或引入语义对齐，但生成式推荐并不等同于“调用一个通用大模型”。

HSTU 更强调将推荐重述为序列转导，并研究可扩展的序列编码结构，不能简单等同于“逐 Token 生成语义 ID”。见 HSTU 原论文。

相对级联架构，语义 ID 生成路线的潜在优势包括统一序列建模、减少多阶段目标不一致，以及利用大规模序列数据扩展模型；主要代价包括 Tokenizer 设计困难、无效生成、训练与服务成本高，以及离线生成指标与在线价值之间的落差。

示例

若一部电影的语义 ID 为 `[12, 87, 203]`，模型根据用户最近观看序列逐 Token 生成新的 ID，再通过码本映射到电影。若生成的组合不对应任何真实电影，则必须判为无效，而不能直接展示。

常见误区与注意事项

- **生成流畅不等于推荐有效。** 语言或序列概率高，不代表用户满意度、转化或长期价值高。
- **语义 ID 可能损失细粒度信息。** 两个相似物品可能共享前缀，但商品规格、库存或时效差异仍可能关键。
- **必须处理无效 ID 与约束。** 生成空间通常大于有效物品集合，需要受限解码和后处理。
- **评估协议尚未统一。** 生成式模型与传统召回模型的候选集合、负样本和指标可能不同，不能只看论文中的单一数值。
- **不是所有业务都需要生成式推荐。** 数据规模较小、延迟敏感或可解释性要求高的系统，成熟级联架构通常更稳妥。
- **前沿结论具有时效性。** HSTU、OneRec 等方法仍在快速演进，应以原始论文、公开代码和可复现实验为准。

16.5 模型谱系速查

模型族	代表方法	主要输入	主要优势	主要局限
邻域协同过滤	UserCF、ItemCF、Swing	用户—物品交互	简单、便于追溯、无需梯度训练但需统计相似度	冷启动、热门偏置
隐向量模型	MF、BPR-MF、SVD++	交互与偏置	表达潜在兴趣、易实现	主要建模二阶关系
特征交互模型	LR、FM、GBDT、DeepFM	稀疏特征与交叉	适合 CTR、表格特征强	依赖特征与采样
双塔召回	DSSM、YouTubeDNN、双塔	用户、物品特征	可预计算、支持 ANN	交叉能力有限
深度排序	Wide & Deep、DIN、DIEN	特征与行为序列	表达复杂兴趣	成本高、易过拟合
序列推荐	GRU4Rec、SASRec	有序行为序列	捕捉兴趣演化	依赖序列质量
图推荐	NGCF、LightGCN、PinSage	用户—物品图	传播高阶协同信号	过平滑、图陈旧
多任务/多场景	MMoE、PLE、ESMM	多目标或多场景样本	共享知识、提升效率	负迁移、权重难调
生成式推荐与序列转导	TIGER、OneRec、HSTU 等不同路线	行为、物品 ID/Token 等，具体按方法区分	序列建模与扩展潜力	不能把不同训练与解码方式混为一谈

本章小结

- LightGCN 通过归一化邻居聚合传播协同信号，再对各层表示加权组合；层数过大会过平滑。
- 混合推荐组合协同、内容、规则和知识信号；多场景建模通过共享专家与场景门控平衡通用性和差异性。
- 语义 ID 生成路线把下一物品预测转化为 Token 序列生成，需处理编码与合法性；HSTU 等其他路线应分别理解。
- 进阶方法不是普遍更优。选择时应比较数据规模、延迟、可解释性、工程成本和业务收益。

关键术语：LightGCN、混合推荐、知识型推荐、多场景建模、生成式推荐、语义 ID、约束解码。

16.6 习题

1. LightGCN 为什么去掉特征变换和非线性激活？这是否意味着图卷积没有必要？
2. 某租房产品使用“用户下单后填写的通勤时间”作为排序特征，可能有什么问题？
3. 多场景模型总体 AUC 提升，但搜索场景下降，应如何排查和改进？
4. 生成式推荐模型生成了一个不在物品库中的语义 ID，应如何处理？这反映了什么设计问题？

> **▼ 参考答案**
>
> 1. 原论文的实验表明这两类操作对协同过滤贡献有限，去掉后可简化模型并改善训练；但邻居聚合仍然保留，高阶协同传播依然是核心，因此不等于图结构无用。
> 2. 该特征发生在下单之后，预测下单前排序时不可得，属于未来信息泄漏；线上无法复现，离线指标会虚高。
> 3. 应分场景查看样本量、标签分布、特征覆盖和损失权重；检查共享参数是否被大场景主导、是否存在负迁移，并尝试场景特定专家、损失加权或梯度平衡，同时设置分场景护栏指标。
> 4. 应判为无效结果并重新解码或回退到合法候选；这反映生成空间与有效物品空间不一致，需要约束解码、合法 ID 校验和兜底策略。

附录 A　核心知识自测清单

先检查当前读过的章节对应条目即可；序列、多目标、在线实验和第 16 章条目留到选学之后。不要把整张清单当作开始做项目的门槛。

- [ ] 能解释召回、排序和重排的目标差异。
- [ ] 能区分分类、回归、Pointwise、Pairwise 和 Listwise。
- [ ] 能从曝光日志构造不含未来泄漏的训练样本。
- [ ] 能解释显式反馈与隐式反馈的标签差异。
- [ ] 能计算 Precision@K、Recall@K，并解释 NDCG@K 的位置折损。
- [ ] 能实现热门推荐、ItemCF 和简单矩阵分解基线。
- [ ] 能解释 BPR 的正物品、负物品和成对目标。

- [ ] 能说明 LR、FM、树模型在 CTR 预估中的特点。
- [ ] 能解释双塔为何适合召回，以及负采样带来的影响。
- [ ] 能说明 Wide & Deep、DeepFM 和候选感知注意力的基本思想。
- [ ] 能构造无未来泄漏的下一物品序列样本。
- [ ] 能区分排序质量、概率校准和多目标权衡。
- [ ] 能提出至少两种新用户和两种新物品冷启动方案。
- [ ] 能解释位置偏差、流行度偏差和反馈回路。
- [ ] 能设计包含护栏指标的 A/B 测试，而不是只看 CTR。
- [ ] 能说明 LightGCN 的邻居聚合、层组合和过平滑风险。
- [ ] 能为冷启动场景设计内容、规则与协同信号的混合策略。
- [ ] 能解释多场景建模中共享参数与场景特异参数的作用。
- [ ] 能区分级联式判别推荐与生成式推荐的建模目标及适用条件。

附录 B　按完成情况推进的学习计划

如果每周能投入 5～8 小时，可以把每一行当作一轮任务，有的用一周，有的需要几周。先完成前面三个阶段，后面的分支二选一，再按兴趣继续。完成产出比赶日期重要。

阶段	学习重点	实践产出
起步	第 1 章、第 2 章直觉与第 5.3 节	跑通小例子，解释一次推荐；公式先不全推
整理数据	第 3～4 章与第 15 章入门步骤	读懂 MovieLens 字段，固定时间划分和评估规则
完成首个项目	第 5 章与第 15 章入门版	比较热门推荐与 ItemCF，写一份小报告；到这里可先停
学习可训练模型	补第 2 章，再读第 6 章	实现 MF 或 BPR-MF，与已有基线比较
分支一：点击预测	第 7 章，按需读第 9 章	有曝光数据后比较 LR 与一个特征交互模型
分支二：向量召回	第 8 章	实现简化双塔，先精确检索，再考虑 ANN
按需进阶	第 10～12 章选一个主题	构造序列样本，或分析排序损失/校准；不要求同时完成
改善已有项目	第 13～14 章与第 15 章进阶版	只加一个变化，例如内容兜底或类别打散，再比较结果
后续选学（不限定一周）	第 16 章	先解释一种方法的输入输出，再按环境选择小规模实验

进阶建议：完成主线后，优先选择一项与当前工作最相关的专题深入，不建议同时学习图推荐、多场景和生成式推荐。若目标是求职或工程落地，先补强第 3～9 章与第 15 章的可复现实验；若目标是研究，再进入第 16 章和原始论文。

附录 C　一句话知识索引

- **协同过滤**：用群体行为发现相似性。
- **矩阵分解**：把用户和物品压缩到同一潜在空间。
- **FM**：用隐向量内积学习稀疏特征交互。
- **双塔**：分别编码用户和物品，以便大规模向量召回。
- **Wide & Deep**：同时建模记忆与泛化。
- **序列推荐**：利用行为顺序预测后续兴趣。
- **Pairwise**：优化物品之间的相对顺序。
- **多任务**：共享表示并同时优化多个目标。
- **探索利用**：在短期收益与获取新信息之间权衡。
- **重排**：从单物品打分走向列表整体优化。
- **A/B 测试**：用受控在线实验估计策略的真实影响。
- **图推荐**：在用户—物品图上传播多跳协同信号。
- **混合推荐**：按场景组合协同、内容、规则与知识信号。
- **多场景建模**：让多个场景借用共同知识，也能表达各自差异，可先用场景特征做简单基线。
- **生成式推荐**：用生成或序列转导思路建模推荐；语义 ID 生成是其中一条路线。

附录 D　参考资料、章节映射与原始论文

D.1 核心资料

编号	资料	版本或范围	链接
[1]	周志华《机器学习》（西瓜书）	清华大学出版社，2016，ISBN 978-7-302-42328-7	通过正规渠道获取；先查第 2、3、5 章，其余按主题选读
[2]	Datawhale 南瓜书（PumpkinBook）	西瓜书公式推导解析，持续更新	在线阅读 · GitHub
[3]	Datawhale fun-rec《深度推荐算法实践（小麦书）》	文档标识 v1.0.0；项目仍在开发，章节可能变化	在线阅读 · GitHub
[4]	Charu C. Aggarwal, *Recommender Systems: The Textbook*	Springer，2016，13 章	Springer · DOI: 10.1007/978-3-319-29659-3

资料定位说明：

- 西瓜书不是推荐系统专著，不覆盖协同过滤、矩阵分解和 CTR 模型；它的价值在于补齐机器学习理论。
- 南瓜书以西瓜书为前提，适合查公式推导，不适合作为独立入门教材。
- fun-rec 的当前版本已经从传统级联架构扩展到生成式范式。其第 1 章对应本指南第 1 章；第 2 章对应第 5～8 章；第 3 章对应第 7～12 章；第 4 章对应第 14 章；第 5～9 章对应本指南第 16.4 节；第 10 章对应工程实践。

- Aggarwal 教材覆盖面广，包含基于内容、知识型、混合、上下文、社交、攻击鲁棒性和高级专题，适合按问题检索。

D.2 本指南与资料章节映射

本指南章节	西瓜书 / 南瓜书	fun-rec（小麦书）	Aggarwal 教材
第 1～4 章	西瓜书第 1～3、11 章；重点第 2 章模型评估	第 1 章；2.1；3.1～3.2；第 10 章	第 1、7 章
第 5～7 章	西瓜书第 3、8、9 章	2.1～2.2；3.1～3.2	第 2～6 章
第 8～10 章	西瓜书第 5、10 章	2.3～2.4；3.1～3.3	第 3、13 章
第 11～12 章	西瓜书第 2、8 章	3.2；3.4	第 13 章
第 13～14 章	西瓜书第 9、16 章	2.5；第 4 章	第 8、13 章
第 15 章	西瓜书第 2 章评估部分	第 10 章	第 7 章
第 16 章	西瓜书第 5、9、10、16 章	2.2～2.5；第 4～9 章	第 10～13 章

说明：表中的章节对应的是知识主题，不要求逐页阅读。例如西瓜书第 16 章强化学习只用于理解状态、动作、奖励和长期回报，不覆盖推荐系统特有的离策略评估问题。

D.3 建议阅读的原始论文

以下是进一步核对模型定义和训练目标的原始论文入口。初次阅读无需全部读完，本次重点复核范围见附录 E：

1. Koren, Y., Bell, R., Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8), 2009. DOI: 10.1109/MC.2009.263。
2. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. *UAI*, 2009. arXiv:1205.2618。
3. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S. Neural Collaborative Filtering. *WWW*, 2017. arXiv:1708.05031。
4. Cheng, H.-T., et al. Wide & Deep Learning for Recommender Systems. *DLRS@RecSys*, 2016. arXiv:1606.07792。
5. Guo, H., Tang, R., Ye, Y., Li, Z., He, X. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *IJCAI*, 2017. arXiv:1703.04247。
6. Zhou, G., et al. Deep Interest Network for Click-Through Rate Prediction. *KDD*, 2018. arXiv:1706.06978。
7. Kang, W.-C., McAuley, J. Self-Attentive Sequential Recommendation. *ICDM*, 2018. arXiv:1808.09781。
8. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *SIGIR*, 2020. arXiv:2002.02126。

阅读论文时建议固定回答五个问题：

1. 它解决哪一个环节的问题，输入和输出是什么？
2. 训练样本与标签如何构造，负样本从哪里来？
3. 损失函数在优化什么，与业务指标有什么差距？
4. 相比基线，提升来自模型结构、特征还是实验设置？
5. 方法在冷启动、稀疏数据、延迟和规模上的限制是什么？

D.4 公开数据集

数据集	内容与规模	适合练习	主要注意事项
MovieLens	电影评分与标签；100K 适合教学，32M 为稳定基准	显式反馈、矩阵分解、排序评估	评分不是曝光日志；Latest 版本会更新，不适合作为可复现研究基准
Amazon Reviews 2023	5.7 亿级评论、商品元数据、用户—商品关系，含时间戳与标准划分	序列推荐、文本与多模态特征、冷启动	评论不等于曝光；需去重、按时间切分，并遵守数据许可
Criteo 1TB Click Logs	24 天展示广告日志，13 个整数特征、26 个哈希类别特征	CTR 预估、特征工程、时间切分	正负样本以不同比例采样，预测概率不能直接当真实 CTR；计算成本高，建议先抽样
KuaiRec	快手短视频交互；小矩阵密度 99.6%，接近完全观测	离线策略评估、无偏推荐、强化学习	单一平台与领域；观看比例不等于长期满意度

数据集使用注意事项：

- 先阅读 README、许可和数据卡，不擅自重新分发受限数据。
- 保留原始时间戳，先按时间划分，再构造特征。
- 明确曝光、交互和标签的定义，不要把“未交互”直接视为负样本。
- 记录预处理、过滤和采样规则；这些规则会显著改变 Recall@K 等指标。
- 公开数据集上的结论只说明在该数据和协议下有效，不应直接外推到生产系统。

附录 E　本次修订与查证范围

本版于 2026-10-30 修订，重点处理容易影响第一次实验的概念和操作：

- 区分评分、曝光点击与未交互采样负例；补全观察窗口、全局时间切分、候选目录和过滤规则。
- 区分 AUC 与概率校准、召回阶段与 Recall 指标、ANN 近邻召回与推荐命中。
- 依据 DIN 原论文第 4.3 节区分 Softmax 注意力示例与 DIN 不强制归一化的局部激活。
- 依据 HSTU 原论文区分序列转导与语义 ID 自回归生成；不把所有生成式方法画成同一种结构。
- 依据 MovieLens 100K 官方 README补充文件字段与评分含义；依据 Criteo 官方说明保留正负样本采样的限制。
- 第 5.3 节是可运行的教学小例子，第 15 章是待执行的真实数据实验步骤。示例的预期输出不代表 MovieLens 上的实验成绩，也不代表线上效果。

附录 D 的教材映射用于按主题查阅，不保证与每个后续网页版本的章节编号一致。开始实践时记录具体仓库提交、数据版本和依赖；本次没有安装并运行全部外部框架，也不声称逐篇复现参考论文。

附录 F　核心术语表

术语	英文或缩写	简明定义	参见
推荐系统	Recommender System	根据用户、物品和上下文，从大量候选中选择并排序内容的系统	第 1 章
召回	Recall / Matching	从全量物品中快速缩小候选集合的阶段，强调覆盖率和效率	第 1、8 章
排序	Ranking	对召回候选进行精细打分并决定顺序的阶段，强调个体相关性	第 1、7、9 章
重排	Re-ranking	在精排之后优化整个列表的去重、多样性、新鲜度和约束	第 1、14 章
曝光	Impression / Exposure	物品实际被展示给用户的一次机会；未曝光的“未点击”不能当普通负样本	第 3 章
显式反馈	Explicit Feedback	用户主动给出的评分、点赞等偏好信号	第 1、6 章
隐式反馈	Implicit Feedback	点击、购买、播放等行为信号；缺少负反馈，且受曝光影响	第 1、6 章
点击率	CTR, Click-Through Rate	曝光后发生点击的条件概率 $P(y=1 \mid u, i, c)$	第 7 章
AUC	Area Under ROC Curve	随机取一正一负样本，正样本得分高于负样本的概率	第 4 章
对数损失	LogLoss	二分类交叉熵的平均值，衡量概率预测质量	第 2、4 章
精确率 at K	Precision@K	前 K 个结果中相关物品所占比例，分母为 K	第 4 章
召回率 at K	Recall@K	前 K 个结果覆盖了多少相关物品，分母为相关物品总数	第 4 章
归一化折损累计增益	NDCG@K	同时考虑相关性和位置的排序指标，越靠前的结果权重越高	第 4 章
协同过滤	Collaborative Filtering	利用群体交互行为寻找相似用户或相似物品的推荐方法	第 5 章
用户协同过滤	UserCF	根据相似用户的历史偏好给当前用户推荐物品	第 5 章
物品协同过滤	ItemCF	根据物品之间的共现或行为相似性给用户推荐相似物品	第 5 章

术语	英文或缩写	简明定义	参见
矩阵分解	Matrix Factorization, MF	把用户和物品映射到同一隐空间，用向量内积表示偏好	第 6 章
贝叶斯个性化排序	BPR	用正物品得分高于负物品的成对目标学习隐式反馈排序	第 6 章
嵌入表示	Embedding	把离散 ID 或特征映射为可学习的低维稠密向量	第 8 章
双塔模型	Two-Tower Model	分别编码用户和物品，输出同维向量并用点积或余弦匹配的召回结构	第 8 章
负采样	Negative Sampling	从大量未交互物品中抽取训练负例，以构造可学习的对比信号	第 6、8 章
近似最近邻	ANN, Approximate Nearest Neighbor	以少量精度损失换取大规模低延迟向量检索的方法	第 8 章
因子分解机	FM, Factorization Machine	用特征隐向量内积建模二阶特征交互的模型	第 7 章
特征交叉	Feature Interaction	组合两个或多个特征形成新信号，例如“用户偏好类别 × 物品类别”	第 7、9 章
Wide & Deep	Wide & Deep Learning	联合训练线性记忆部分和深度泛化部分的排序模型	第 9 章
DeepFM	Deep Factorization Machine	共享 Embedding，同时用 FM 和深度网络建模低阶、高阶交互	第 9 章
深度兴趣网络	DIN, Deep Interest Network	用候选感知注意力对用户历史行为动态加权的 CTR 模型	第 9 章
序列推荐	Sequential Recommendation	根据有序行为序列预测下一个或下一批交互	第 10 章
学习排序	LTR, Learning to Rank	直接优化物品相对顺序或整个列表的排序方法族	第 11 章
多任务学习	Multi-task Learning	共享部分参数，同时优化多个相关目标的建模方式	第 12 章
概率校准	Probability Calibration	使预测概率与真实发生率一致的调整过程	第 12 章
冷启动	Cold Start	新用户、新物品或新系统缺少历史数据，难以估计偏好的问题	第 13 章
探索与利用	Exploration-Exploitation	在短期使用当前最优策略与尝试不确定选项获取信息之间权衡	第 13 章
位置偏差	Position Bias	靠前位置更容易被点击，导致点击标签混入展示位置影响	第 14 章

术语	英文或缩写	简明定义	参见
A/B 测试	A/B Testing	将流量随机分配给不同策略，以估计策略真实影响的在线实验	第 14 章
图推荐	Graph Recommendation	在用户—物品图上传播邻居信息以学习高阶协同关系的推荐方法	第 16 章
多场景建模	Multi-Scenario Modeling	利用多个场景的数据与差异，可从共享模型加场景特征开始	第 16 章
生成式推荐	Generative Recommendation	一类用生成或序列转导思路建模推荐的方法，语义 ID 自回归生成是其中一条路线	第 16 章