

多角度审视推荐系统

Question:

1. 推荐系统如何选取和处理特征?
2. 推荐系统在召回策略上有什么?
3. 热门冷门体现在哪? 如何提高冷门推荐?
4. 如何根据用户的兴趣构建推荐模型和优化目标?
5. 如何让用户对推荐结果进行反馈?
6. 如何解决用户冷启动问题?
7. 什么是: 保真与召回(CER)? 如何平衡解决方法?

Question: 1. 推荐系统如何选取和处理特征?

推荐系统的特征工程

- (1) 哪些特征工程是应该遵循的原则?
 - (1) 特征是常用的特征吗?
 - (2) 特征在特征的特征上进行处理,生成可读推荐系统训练和推理
时用的特征向量?
- (1) 尽可能保留推荐环节用户行为过程中所有有用信息,尽量避免冗余信息
- (2)
 - ① 用户行为数据 {显式行为
隐式行为 Δ }
 - ② 用户关系数据 {强关系(朋友/关注...)
弱关系(地区/有二度关系...)
- ③ 属性-标签类数据
 $\text{用户} \rightarrow \begin{cases} \text{人口统计学特征}/age/location\cdots \\ \text{兴趣标签} \end{cases}$
- 物品 $\rightarrow \begin{cases} \text{物品特征} \\ \text{物品属性}/type/price/rating/\cdots \end{cases}$
- ④ 内容类数据
语言 ...

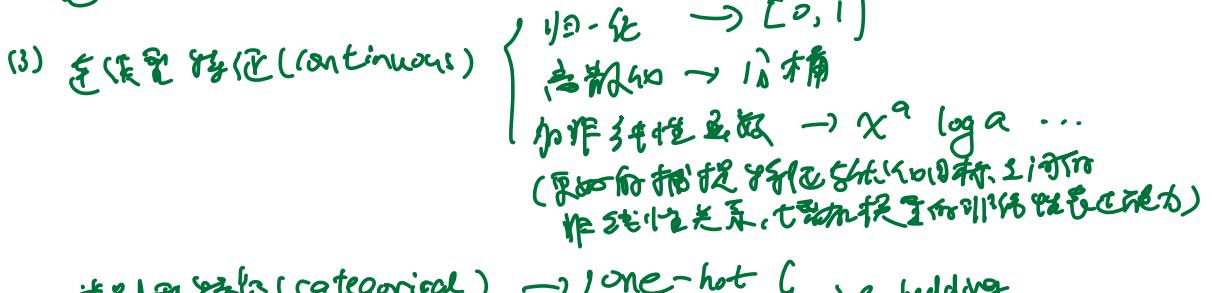
⑤ 上下文信息 (context)

时间、地点、节假日、历史...

⑥ 统计特征

平均 CTR、物品热冷程度...

⑦ 组合类特征 (通过 model 内部组合)



从这个意义上讲，传统的人工特征组合、过滤的工作已经不存在了，取而代之的是将特征工程与模型结构统一思考、整体建模的深度学习模式。不变的是，只有深入了解业务的运行模式，了解用户在业务场景下的思考方式和行为动机，才能精确地抽取出最有价值的特征，构建成功的深度学习模型。

Question 2. 推荐系统召回策略主要有哪些？

1. 召回层后端

待召回候选量大、速度快、模型简单、特征较少，尽量让用户感兴趣
待召回候选量大、速度快、模型简单、特征较少，尽量让用户感兴趣
待召回候选量大、速度快、模型简单、特征较少，尽量让用户感兴趣

2. 策略 多路召回

多路召回：采用不同的策略，特征丰富度强，同时召回一部分候选集，
召回率提升集中在一定区间内推荐模型使用。

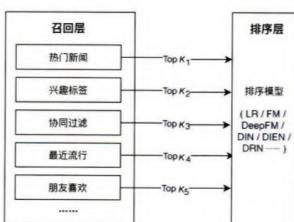


图 5-5 常见的多路召回策略

事实上，召回策略的选择与业务强相关。对视频推荐来说，召回策略可以是“热门视频”“导演召回”“演员召回”“最近上映”“流行趋势”“类型召回”等。

每一路召回策略会拉回 K 个候选物品，对于不同的召回策略， K 值可以选择不同的大小。这里的 K 值是超参数，一般需要通过离线评估加线上 A/B 测试的方式确定合理的取值范围。

embedding: Embedding + 向量的 hash (Youtube)

Question 3. 3. 模型实时性体现在哪？如何提高实时性？

类型：

① **模型实时性：**

实时地收集和更新推荐模型的输入特征，
保证推荐系统能根据最新的特征进行计算。

影响模型实时性的因素

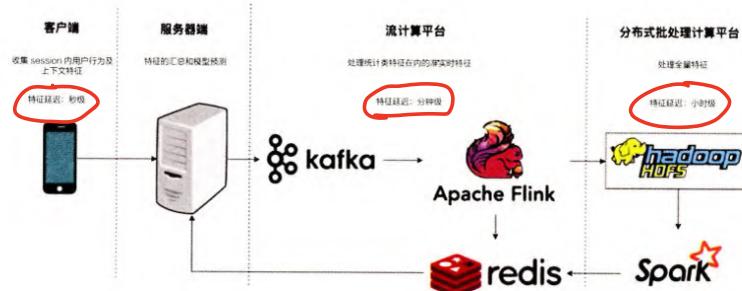


图 5-7 推荐系统的数据流架构图

1. 客户端实时性

如果客户端能够缓存 session 内部的行为，将其作为与上下文特征同样的实时特征传给推荐服务器，那么推荐模型就能够实时地得到 session 内部的行为特征，进行实时的推荐。这就是利用客户端实时特征进行实时推荐的优势所在。

2. 流计算平台的准实时特征处理

3. 分布式批处理平台的全量特征处理

用途
1. 模型训练与高阶评估
2. 异常检测与故障恢复，依赖特征使用

(长延迟, 不实时)

② **模型实时性：**希望更快速地抓住全局层面的数据变化模式，发现新的趋势和相关性。

以某电商网站“双 11”的大量促销活动为例，特征的实时性会根据用户最近的行为更快地发现用户可能感兴趣的的商品，但绝对不会发现一个刚刚流行起来的爆款商品、一个刚刚开始的促销活动，以及与该用户相似的人群最新的偏好。要发现这类全局性的数据变化，需要实时地更新推荐模型。

模型实时性与训练方式相关：

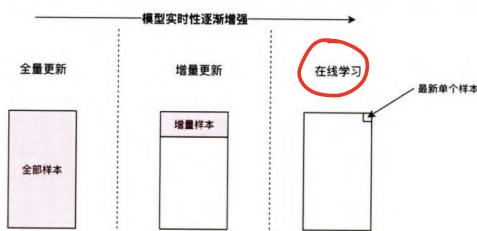


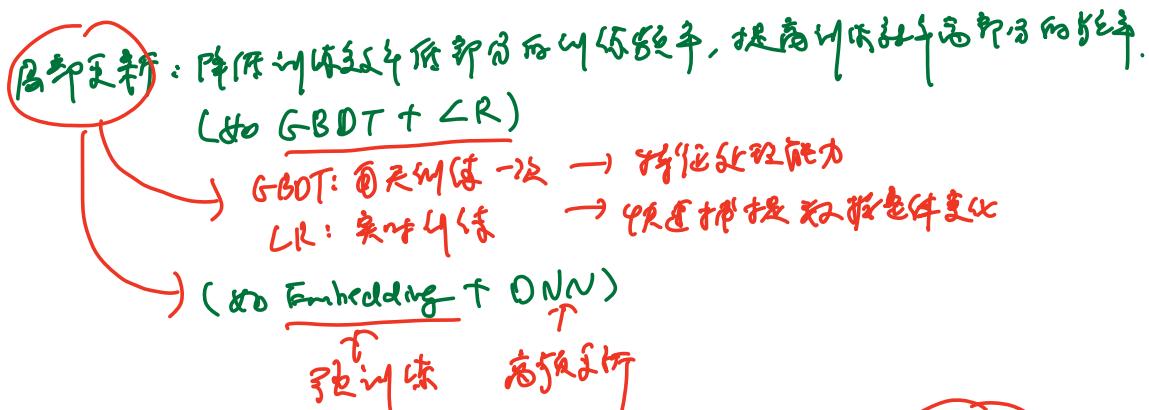
图 5-8 模型的实时性与训练方式的关系

在线学习(Online Learning)(祝愿精神)

在线学习是进行模型实时更新的主要方法，也就是在获得一个新的样本的同时更新模型。与增量更新一样，在线学习在技术上也通过SGD的训练方式实现，但由于需要在线环境进行模型的训练和大量模型相关参数的更新和存储，工程上的要求相对比较高。

在线学习的另一个附带问题是模型的稀疏性不强，例如，在一个输入特征向量达到几百万维的模型中，如果模型的稀疏性好，就可以在模型效果不受影响的前提下，仅让极小一部分特征对应的权重非零，从而让上线的模型体积很小（因为可以摒弃所有权重为0的特征），这有利于加快整个模型服务的过程。但如果使用SGD的方式进行模型更新，相比batch的方式，容易产生大量小权重的特征，这就增大了模型体积，从而增大模型部署和更新的难度。为了在线学习过程中兼顾训练效果和模型稀疏性，有大量相关的研究，最著名的包括微软的FOBOS^[1]、谷歌的FTRL^[2]等。

在线学习的另一个方向是将强化学习与推荐系统结合，在3.10节介绍的强化学习推荐模型DRN中，应用了一种竞争梯度下降算法，它通过“随机探索新的深度学习模型参数，并根据实时效果反馈进行参数调整”的方法进行在线学习，这是在强化学习框架下提高模型实时性的有效尝试。



Question 4. 如何根据具体场景构建推荐模型的优化目标？

YouTube - 以观看时长为优化目标。

根本原因：广告插在视频中间，播放时间越长，用户的广告越多。

Alibaba - 模型优化与应用场景的统一性 (ESMM) (译者 Paper)

Question 5 除了moder. 有没有其他更直接影响推荐系统的因素？“揣摩人心”

5.5.2 Netflix 对用户行为的观察

Netflix 是美国最大的流媒体公司，其推荐系统会根据用户的喜好生成电影的推荐列表。除了影片的排序，最能影响点击率的因素其实是影片的海报预览图。举例来说，一位喜欢马丁·达蒙（美国著名男明星）的用户，当看到影片的海报上有马丁·达蒙的头像时，点击该影片的概率会大幅增加。Netflix 的数据科学家在通过 A/B 测试验证这一点后，着手开始对影片预览图的生成进行优化（如图 5-13 所示），以提高推荐结果整体的点击率。



图 5-13 Netflix 不同预览图的模板

在具体的优化过程中，模型会根据不同用户的喜好，使用不同的影片预览图模板，填充以不同的前景、背景、文字等。通过使用简单的线性“探索与利用”（Exploration&Exploitation）模型验证哪种组合才是最适合某类用户的个性化海报。

在这个问题中，Netflix 并没有使用复杂的模型，但 CTR 提升的效果是 10% 量级的，远远超过改进推荐模型结构带来的收益。这才是从用户和场景出发解决问题。这也符合 5.3 节提出的“木桶理论”的思想，对推荐系统效果的改进，最有效的方法不是执着地改进那块已经很长的木板，而是发现那块最短的木板，提高整体的效果。

5.5.3 观察用户行为，在模型中加入有价值的用户信息

再举一个例子，图 5-14 是美国最大的 Smart TV（智能电视）平台 Roku 的主页，每一行是一个类型的影片。但对一个新用户来说，系统非常缺少关于他的点击和播放样本。那么对 Roku 的工程师来说，能否找到其他有价值的信息来解决数据稀疏问题呢？



图 5-14 捕捉包含关键信息的用户行为

这就要求我们回到产品中，从用户的角度理解这个问题，发现有价值的信号。针对该用户界面来说，如果用户对某个类型的影片感兴趣，则必然会向右滑动鼠标或者遥控器（如图 5-14 中红色箭头所指），浏览这个类型下其他影片，这个滑动的动作很好地反映了用户对某类型影片的兴趣。

引入这个动作，无疑对构建用户兴趣向量、解决数据稀疏问题，进而提高推荐系统的效果有正面的作用。广义上讲，引入新的有价值信息相当于为推荐系统增加新的“水源”，而改进模型结构则是对已有“水源”的进一步挖掘。通常，新水源带来的收益更高，开拓难度却小于对已有水源的持续挖掘。

Question 6 6. 如何解决冷启动问题？

冷启动 {
 用户注册数
 物品
 系统

策略：

1. 在于规则的冷启动过程，如 Airbnb

Airbnb 是全球最大的短租房中介平台。在新上线短租房时，Airbnb 会根据该房屋的属性对该短租房指定一个“聚类”，位于同样“聚类”中的房屋会有类似的推荐规则。那么，为冷启动短租房指定“聚类”所依靠的规则有如下三条：

- (1) 同样的价格范围。
- (2) 相似的房屋属性（面积、房间数等）。
- (3) 距目标房源的距离在 10 公里以内。

2. 丰富冷启动过程中可获得的用户和物品特征，如

①. 用户注册信息

②. 第三方 DMP (Data Management Platform)

③. 物品内容特征

④. 引导用户定义的冷启动特征

3. 利用强化学习、迁移学习和 EE 机制

Question 7 什么是探索与利用 (EE)？如何解决方法？

1. 传统的探索 (E) 方法 {
 ε-Greedy
 Thompson Sampling
 UCB

2. 个性化 EE 方法

3. 基于模型的 EE 方法

(i) 多臂老虎机 (Multi-armed Bandit Problem)

• 假设每个商品不一样权重，针对用户、针对叶尼塞机解决问题

(ii) ε-Greedy 算法

• 以 ε 科元率选取新的老虎机；从概率中选出当前老虎机收益最大的。

(2) Thompson Sampling

Thompson Sampling^[8]是一种经典的启发式探索与利用算法。该算法假设每个老虎机能够赢钱（这里假设赢钱的数额一致）的概率是 p ，同时概率 p 的分布符合 beta(win, lose) 分布，每个老虎机都维护一组 beta 分布的参数，即 win, lose。每次试验后，选中一个老虎机，摇臂后，有收益（这里假设收益是二值的，0 或 1）则该老虎机的 win 参数增加 1，否则该老虎机的 lose 参数增加 1。

每次选择老虎机的方式是：利用每个老虎机现有的 beta 分布产生一个随机数 b ，逐一生成每个老虎机的随机数，选择随机数中最大的那个老虎机进行尝试。

. Thompson Sampling 对新产品向性

(3) UCB 算法

UCB 是经典的启发式探索与利用算法。与 Thompson Sampling 算法的理论基础一样，二者都利用了分布的不确定性作为探索强弱程度的依据。但在形式上，UCB 更便于工程实现，其算法流程如下：

(1) 假设有 K 个老虎机，对每个老虎机进行随机摇臂 m 次，获得老虎机 j 收益的初始化经验期望 \bar{x}_j 。

(2) 用 t 表示至今摇臂的总次数，用 n_j 表示第 j 个老虎机至今被摇臂的次数，计算每个老虎机的 UCB 值：

$$\text{USB}(j) = \bar{x}_j + \sqrt{\frac{2\log t}{n_j}} \quad (\text{式 5-3})$$

$\bar{x}_j \rightarrow$ 老虎机 j 之前实验的收益期望 (利用)

$\sqrt{\frac{2\log t}{n_j}} \rightarrow$ 置信区间宽度 (探索)

(3) 选择 UCB 值最大的老虎机 i 摆臂，并观察其收益 $X_{i,t}$ 。

(4) 根据 $X_{i,t}$ 更新老虎机 i 的收益期望值 \bar{x}_i 。

(5) 重复第 2 步。

Thompson Sampling & UCB 都用，但无法引入个性化特征。

(2) 个性化向量方法

LinUCB

(3) 深度学习方法

DRN

(4) EE 的应用场景

1. 物品冷启动
2. 新用户推荐
3. 侧边栏结果多样性