# 多角度审视推荐系统

原创 子墨 搜索与推荐Wiki 2020-11-09

收录于话题
#推荐相关笔记

32个



**精彩推荐 ▼**



编辑：子墨为客
来源：《深度学习推荐系统》笔记，并进行补充和说明

## 推荐系统中的特征工程

特征的本质其实是对某个行为过程相关信息的抽象表达。

推荐系统中的常用特征：

### 1、用户行为数据

包括显性反馈数据（explicit feedback）和隐形反馈数据（implicit feedback）。

用户行为类特征得两种处理方式：

- 将代表用户行为的物品id序列转换成multi-hot编码，将其作为特征向量
- 预先训练好物品的Embedding向量，再通过平均或者类似于DIN模型注意力机制的方法生成历史行为的Embedding向量，将其作为特征向量。

### 2、用户关系数据

用户关系数据可以分为「显性」和「隐性」，或者「强关系」和「弱关系」。

### 3、属性、标签类数据

描述用户或者物品的特征，属性和标签的主体可以是用户、也可以是物品。

成熟得公司往往会建立一套用户和物品的标签体系，由专门的团队负责维护，典型的比如电商公司的商品分类体系。

### 4、内容类数据

内容类数据也是描述物品或者用户的数据，但无法直接转换成推荐系统可以消化的特征，可以通过自然语言处理、计算机视觉技术提取内容的关键内容特征，再输入到推荐系统。

### 5、上下文消息

上下文信息是描述推荐行为产生时的场景信息，比如：时间、地点、季节、月份等。

「引入上下文信息的目的是尽可能的保存推荐行为发生场景的信息」，如果不引入上下文特征，则推荐系统无法捕获到与这些场景有关的有价值的信息。

### 6、统计类特征

统计类特征是指通过统计方法计算出的特征，比如CTR、CVR、物品热门程度、物品流行度等。

统计类特征往往与最后的预测目标有较强的相关性，因此是绝不应该被忽视的重要特征类别。

### 7、组合类特征

组合类特征是指将不同特征进行组合后生成的新特征。比如年龄+性别组成的人口属性分段特征。

在早期的推荐系统中，模型不具有特征组合的能力，但是随着更多深度学习推荐系统的提出，组合类特征不一定通过人工组合、人工筛选的方法选出，还可以交给模型进行自动处理

常用的特征处理方法：

### 1、连续类特征

最常见的处理手段包括：归一化、离散化、加非线性函数等方法

归一化的目的是：统一各特征的量纲

离散化的目的是：防止连续值带来的过拟合现象及特征值分布不均匀的情况

加非线性函数是把原来的特征通过非线性函数做变换，然后把原来的特征及变换后的特征一起加入模型进行训练的过程，常用的非线性函数包括：

$x^a, log(a), log(\frac{x}{1-x})$

加入非线性函数的目的是更好的捕获特征与优化目标之间的非线性关系，增强模型的非线性表达能力

### 2、类别型特征

常见的处理方法是：multi-hot编码，但主要问题是特征向量维度过大，特征过于稀疏，容易造成模型欠拟合，模型的权重参数的数量过大，导致模型收敛过慢。

因此常见的处理方法是先将类别特征编码成稠密的Embedding向量。

## 推荐系统召回层的主要策略

召回层和排序层的特征：

- 召回层：候选集大、速度快、模型简单、特征较少、尽量让用户感兴趣的物品在这个阶段能够被快速召回，即保证相关物品的召回率。
- 排序层：首要目标是得到精准的排序结果，需处理的物品数量少，可利用较多特征，使用比较复杂的模型

### 多路召回策略

所谓的「多路召回策略」是指采用不同的策略、特征或简单模型，分别召回一部分候选集，然后把候选集混合在一起供后续排序模型使用的策略。

可以明显看出，「多路召回策略」是在“计算速度”和“召回率”之间进行权衡的结果。

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

多路召回策略的缺点：

- 是实用的召回方法，但从策略选择到候选集的大小参数的调整都需要人工参与
- 策略之间的信息也是割裂的，无法综合考虑不同策略对一个物品的影响

「基于Embedding的召回方法是一种综合性强且计算速度也能满足需求的召回方法」

### 基于Embedding的召回方法

事实上，多路召回中使用“兴趣标签”、“热门度”、“流行趋势”、“物品属性”等信息都可以作为Embedding召回方法中的附加细腻下（side information）融合进最终的Embedding向量中。

Embedding召回的另一个优势在于评分的连续性。Embedding召回可以把Embedding间的相似度作为唯一的判断标准，因此可以随意限定召回的候选集大小。

矩阵分解、因子分解机等简单模型也完全可以得到用户和物品的Embedding向量。

## 推荐系统的实时性

「模型更新的越频繁，实时性越好，损失越小，效果越好」

推荐系统的实时性：

- 「特征」的实时性：推荐系统的更新速度越快，代表用户最近习惯和爱好的特征更新越快，越能为用户进行更有实时性的推荐
- 「模型」的实时性：推荐系统更新的越快，模型越容易发现最新流行的数据模式，越能让模型快速抓住最新的流行趋势

推荐系统特征实时性的三个点：

- 客户端实时特征
- 流计算平台的准实时特征处理
- 分布式批处理平台的全量特征处理

推荐系统模型的实时性往往是从更全局的角度考虑问题，特征的实时性力图用更准确的特征描述用户、物品和相关场景，从而让推荐系统给出更符合当时场景的推荐结果。而模型的实时性则是希望更快的抓住全局层面的新数据模式，发现新的趋势和相关性。

模型的实时性由弱到强的训练方式分别是全量更新、增量更新和在线学习。

- 全量更新：指利用某时间段内的所有训练样本进行训练
- 增量更新：仅将新加入的样本喂给模型进行增量训练（增量更新的模型往往无法找到全局最优点，因此在实际的推荐系统中，经常采样增量更新与全局更新相结合的方式，在进行了几轮增量更新后，在业务量较小的时间窗口进行全局更新，纠正模型在增量更新过程中积累的误差）
- 在线学习：需要在线上环境进行模型的训练和大量模型相关参数的更新和存储，工程上要求比较高，比较著名的在线学习算法包括：FOBSO、FTRL等
- 局部更新：
    - GBDT+LR：Facebook采取的部署方法是每天训练一次GBDT模型，固定GBDT模型后，实时训练LR模型以快速捕捉数据整体的变化，通过模型的局部更新，做到GBDT和LR能力的权衡

- pre Embedding + 深度学习模型：因为Embedding层参数由于占据了深度学习模型参数的大部分，其训练过程会拖慢模型整体的收敛速度，因此常见的做法是Embedding层单独预训练，Embedding层以上的模型部分高频更新的混合策略

### 客户端模型的更新

更多指的是Embedding相关的操作。

### 用“木桶理论”看待推荐系统的迭代升级

在实际的改进过程中，“抓住一点，重点提升”是工程师应该采取的策略。而准确的找到这个关键点的过程就要求我们以“木桶理论”看待这个问题，找到拖慢推荐系统事实性的最短的那块模板，替换或者改进它，让“推荐系统”这个木桶能够盛下更多的水。

## 如何合理设定推荐系统中的优化目标

如果一项技术本身是新颖的，先进的，但应用方向与实际需求的方向有偏差，那这项技术的成果不可能是显著的。

因此犯战略性的失误，合理设定推荐系统的优化目标是每位推荐工程师在构建推荐系统之前应该着重思考的问题。

设定一个合理的推荐系统优化目标，首先需要确立一个合理的原则，对于一家商业公司而言，在绝大数情况下，推荐系统的目标是完成某个商业目标，所以根据公司的商业目标来制定推荐系统的优化目标理应作为合理的战略性目标。

### Youtube以观看时长为优化目标的合理性

目前推荐场景中大多数的优化目标都是点击率，但是往往观察的指标是通过点击率进一步观察其它的指标，比如电商环境中的交易额，下单次数等；比如视频网站中的观看时长。

Youtube的主要商业模式是广告收入，广告收入是与用户观看时长成正比的，不可否认，点击率等指标与用户播放时长有相关性，但二者之间仍存在一些“优化动机”上的差异。如果以点击率为优化目标，对于一些标题党、封面劲爆的视频比较友好。因此**推荐目标的差异会导致推荐系统倾向性的不同，进而影响到能否完成商业目标**

针对预测用户时长问题，Youtube将其转化为二分类问题，对于神经网络输出层采用加权逻辑回归进行输出。

**电商场景中的商业目标一般是交易额（比如阿里巴巴），这时候就不是简单的CTR模型了，具体的实践可以参考论文：**Entire Space Multi-Task Model

要点：

- 推荐模型的结构不是构建一个好的推荐系统的「银弹」，真正的「银弹」是对用户行为和应用场景的观察
- 从用户的角度理解某个行为，继而发现有价值的信号
- 从用户的角度思考问题，构建模型

## 冷启动

冷启动就不多介绍了，可以参考下面的两篇文章：

- 《推荐系统开发实战》之冷启动问题
- EE和冷启动中的多臂老虎机问题