1. 推荐系统技术概览

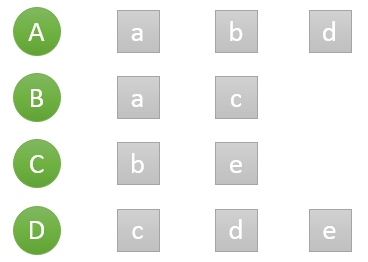
深度学习推荐系统之前

1. 基于用户的协同过滤
2. 原理

构建用户为行坐标，商品为列坐标的共现矩阵，在生成共现矩阵之后，推荐问题就转换为预测矩阵中问号元素的问题（其实是计算用户与用户向量之间的余弦相似度）。

1. 问题

用户数往往大于物品数，所以存储共现矩阵的开销很大；同时用户的数据向量往往会十分稀疏，导致找到相似用户的准确度是非常低的。



1. 基于物品的协同过滤
2. 原理

通过计算共现矩阵中物品列向量的相似度得到物品之间的相似矩阵，因为一个用户购买过物品a和物品b，说明a，b之间有相似性。再找到用户的历史正反馈物品的相似物品进行进一步排序和推荐

1. 问题

共现矩阵的稀疏性，出现推荐结果的头部效应明显。

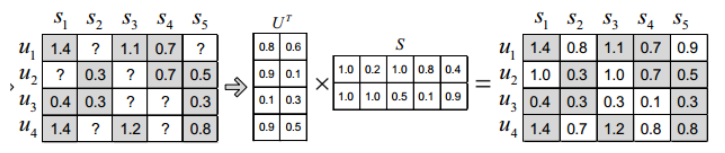


1. 矩阵分解算法 --- 协同过滤的进化，使任意的用户和物品之间都可以得到预测分值。
2. 原理

矩阵分解算法则期望为每一个用户和视频生成一个隐向量，将用户和视频定位到隐向量的表示空间上，距离相近的用户和视频表明兴趣特点接近。在推荐过程中就应该把距离相近的视频推荐给目标用户。矩阵分解算法将m \* n维的共现矩阵分解为m \* k维的用户矩阵U，和k \* n维的物品矩阵V。其中m为用户的数量，k为隐向量的维度。K越小，隐向量的表达能力弱但是泛化能力强。

1. 存在的问题

没有办法加入用户和物品上下文的信息。同时在缺乏用户历史行为时，无法进行有效的推荐。



1. 逻辑回归 --- 融合多种特征的推荐模型
2. 原理

极大似然法更新模型参数，在考虑融合不同特征，形成比较全面的推荐结果。同时逻辑回归的数学意义满足CTR点击事件的实际意义。（点击还是不点击）。

1. 存在的问题

无法进行特征交叉、特征筛选等一系列高级操作。在仅使用单一特征而不是交叉特征进行判断的情况下，有时不仅是信息损失的问题，甚至会得出错误的结论。

从FM到FFM --- 自动特征交叉的解决方案

1. POLY2模型 --- 对特征进行暴力组合
2. 原理

该模型对所有特征进行两两交叉，并对所有的特征组合赋予权重。但POLY2模型本质上仍是线性模型，其训练方法与逻辑回归并无区别

1. 存在的问题

参数的增加，加大了训练的复杂度；该方法使得原本稀疏的向量更为稀疏。

1. FM模型 --- 隐向量特征交叉
2. 原理

隐向量的引入可以很好的解决数据稀疏的问题。在某推荐场景下，样本有两个特征，分别是频道（channel）、品牌（brand），某训练样本的特征组合是（ESPN，Adidas）。在POLY2中只有当ESPN和Adidas都出现时，模型才能学习到这个组合特征的权重，而在FM中ESPN的隐向量也可以通过（ESPN,Gucci）样本进行更新，Adidas的隐向量也可以通过（NBC，Adidas）样本进行更新，这大幅降低了模型对数据稀疏性的要去。设置一个从未出现的特征组合也可以具备隐向量。FM提高了泛化能力。

1. 存在的问题
2. FFM模型 --- 引入特征域的概念
3. 原理
4. 存在的问题
5. GBDT + LR --- 特征工程模型化的开端
6. 原理

利用GBDT（决策树组成的森林）进行特征的选取

1. 存在的问题

深度学习推荐系统之后

1. AutoRec模型

模型从神经网络的角度出发，使用一个单隐层的AutoEncoder泛化用户或物品评分，使模型具有一定的泛化能力。也分为基于物品的AutoRec和基于用户的AutoRec.但是由于AutoRec模型比较简单，使其存在一定的表达能力不足的问题。通过最小化以下公式得到隐向量表达：

模型的输出向量为

1. Deep Crossing

该模型中没有任何人工特征工程的参与，原始特征经Embedding后输入神经网络层，将全部特征交叉的任务交给模型。相比之前FM、FFM模型只具备二阶特征交叉的能力。Deep Crossing模型可以通过调整神经网络的深度进行特征之间的“深度交叉”。

·Embedding层：将稀疏的类别特征转换成稠密的Embedding向量，Embedding层的结构以经典的全连接层结构为主。一般来说Embedding向量的维度应远远小于原始的稀疏特征向量。

·Stacking层：作用比较简单，是把不同的Embedding特征和数值特征拼接在一起，形成新的包含全部特征的特征向量

·Multiple Residual Units层：该层的主要结构是多层感知机

·Scoring层：为了拟合优化目标而存在的。

1. NeuralCF模型 --- CF与深度学习的结合

元素积：向量中对应位置的元素进行相乘。

NeuralCF模型实际上提出一个模型框架，它基于用户向量和物品向量这两个Embedding层，利用不同的互操作层进行特征的交叉组合，并且可以灵活的进行不同互操作层的拼接。从这里可以看出深度学习构架推荐模型的优势，可以灵活地组合不同的特征，按需增加或减少模型的复杂度。

1. 存在的问题

与协同过滤算法一样，并没有引入更多其他的特征

1. PNN模型

引入了乘积层的概念，使用向量内积的方式称为IPNN（Inner Product-based Neural Network）、使用向量外积的方式称为OPNN（Outer Product-based Neural Network）

1. Wide&Deep模型

“记忆能力”可以被理解为模型直接学习并利用历史数据中物品或者特征的“共现频率”的能力。

“泛化能力”可以被理解为模型传递特征的相关性，以及发掘稀疏甚至从未出现过的稀有特征与最终标签相关性的能力。

简单模型的“记忆能力”强，深度神经网络的“泛化能力”强。

Wide&Deep模型把单输入层的Wide部分与由Embedding层和多隐层组成的Deep部分连接起来，一起输入最终的输出层。单层的Wide部分善于处理大量的稀疏id类特征

1. Deep&Cross模型
2. FNN --- 用FM的隐向量完成Embedding层初始化
3. Embeddig技术

Embedding向量能够表达相应对象的某些特征，同时向量之间的距离反映了对象之间的相似性。在大量语料输入的情况下，Embedding技术设置可以挖掘出一些通用知识。既然Embedding能够对“词”进行向量化，那么其他应用领域的物品也可以通过某种方式生成其向量化表示。

例如：对电源进行Embedding的时候，那么Embedding(复仇者联盟)和Embedding(钢铁侠)在Embedding向量空间两点之间的距离就应该很近，而Embedding(复仇者联盟)和Embedding(乱世佳人)的距离会比较远。同理再对商品进行向量化的时候，同种类型的商品之间词向量较为接近，不同类型的商品的向量较远。

1. Word2Vec

参考：<https://www.zhihu.com/question/44832436>

1. Item2Vec

Word2Vec可以对词“序列”中的词进行Embedding，那么对用户购买“序列”中的一个商品，用户观看“序列中”的一个电影，也应该存在相应的Embedding方法，这就是Item2Vec方法。Item2Vec和Word2Vec唯一的不同在于，Item2Vec抛弃了时间窗口的概念，认为序列中任意两个物品都相关，因此在Item2Vec的目标函数中可以看出，其是两两物品的对数概率的和，而不是时间窗口内物品的对数概率的和。

1. Graph Embedding --- 引入更多结构信息的图嵌入技术

Word2Vec和Item2Vec是Embedding技术的基础性方法，但二者都是建立在“序列”样本的基础上。在互联网场景下，数据对象之间更多呈现的是图结构。典型的场景是由用户行为数据生成的数据关系图，以及由属性和实体组成的知识图谱。Graph Embedding是一种对图结构中的节点进行Embedding编码的方法。最终生成的节点Embedding向量一般包含图的结构信息及附近节点的局部相似性信息。

1. DeepWalk --- 基础的Graph Embedding方法

它的主要思想是在物品组成的图结构上进行随机游走，产生大量物品序列，然后将这些物品序列作为训练样本输入Word2Vec进行训练，得到物品Embedding。因此，DeepWalk可以被看做连接序列Embedding和Graph Embedding的过渡方法。

1. Node2Vec --- 同质性和结构性的权衡

“同质性”：指的是距离相近节点的Embedding应尽量近似。DFS搜索

“结构性”：指结构上相似的节点的Embedding应尽量近似。BFS搜索

同质性相同的物品则是各类的同品类、同属性，或者经常被一起购买的商品，而结构性相同的物品则是各品类的爆款、各品类的最佳凑单商品。

1. EGES --- 阿里巴巴的综合性Graph Embedding方法

…

Embedding层将高维向量映射为低纬向量，本身在训练时权重矩阵的参数就十分多，会拖慢整个网络的训练速度。另一方面物品和用户的Embedding向量是比较稳定的，在短时间内不会发生太大的改变，所以Embedding的训练频率不用很高，设置可以降到“周”的级别。但是上层神经网络为了尽快抓住最新的数据整体趋势信息，往往需要高频率训练甚至是实时训练。使用不同的训练频率更新Embedding模型和神经网络模型是训练开销和模型效果二者之间权衡后的最优方案。

1. Embedding作为推荐系统召回层的方法

“召回（match）”：指全量信息集合中触发尽可能多的正确结果，并将结果返回给“排序”。回的方式有多种：协同过滤、主题模型、内容召回和热点召回等。

“排序（rank）”：是对所有召回的内容进行打分排序，选出得分最高的几个结果推荐给用户。

参考：<https://blog.csdn.net/akenseren/article/details/98208341>

1. 多角度审视推荐系统
2. 特征的选取
3. 用户行为类数据

在推荐系统中一部分表现为显性反馈行为（explicit feedback）和隐性反馈行为（implicit feedback）两种。其中显性反馈行为有：对商品的评分、对视频的点赞、对歌曲，歌手的评分等。隐性反馈行为有：点击、加入购物车、评论等。在深度学习的时代，显性反馈行为收集难度大，数据量小。隐性反馈行为显得越来越重要。

1. 用户关系数据

用户行为数据是人与物之间的“连接”日志，那么用户关系数据就是人与人之间连接的记录。

1. 属性、标签类数据

用户属性、物品属性、标签类数据是最重要的描述型特征。

1. 内容性数据

内容性数据可以看作属性标签型特征的延伸，它们同样是描述物品或用户的数据，但相比标签类特征，内容类数据往往是大段的描述型文字、图片、甚至是视频。这一部分特征需要经过自然语言处理、计算机视觉等手段提取关键内容特征，再输入推荐系统。

1. 上下文信息

这是一类描述推荐系统行为产生的场景的信息。最常用的上下文信息是“时间”和通过GPS获得的“地点”信息。也包括时间、地点、季节、月份等信息。这些信息是为了让推荐系统捕捉到与场景相关的信息。

1. 统计类信息

是指通过统计方法计算出的特征，例如历史CTR、历史CVR、物品热门程度、物品流行程度等。统计类特征一般是连续型特征。

1. 存在的问题
2. 位置偏置

用户点击某个广告，并非出于喜好，有可能仅仅与展示位置有关。

1. 选择偏置

物品只在相对靠后的位置、展示给了少量的用户，属于尾部物品。也就是说，选择偏置是推荐系统对长尾物品推荐准确性不高的一个原因。

1. 马太效应

强者恒强，弱者愈弱。

1. 冷启动

推荐系统需要根据用户的历史行为和兴趣预测用户未来的行为和兴趣，因此大量的用户行为数据就称为推荐系统的重要组成部分和先决条件。很多在开始阶段就希望有个性化推荐应用的网站来说，如何在没有大量用户数据的情况下设计个性化推荐系统并且让用户对推荐结果满意从而愿意使用推荐系统，就是冷启动问题。

冷启动问题主要分为3类：

·用户冷启动：用户冷启动主要解决如何给新用户做个性化推荐的问题。 当新用户到来时，没有他的行为数据，所以无法根据他的历史行为预测其兴趣，从而无法借此给他做个性化推荐。

·物品冷启动：物品冷启动主要解决如何将新的物品推荐给可能对它感兴趣的用户这一问题。

·系统冷启动：系统冷启动主要解决如何在一个新开发的网站上(没有用户，也没有用户行为，只有一些物品的信息)设计个性化推荐系统，从而在网站刚发布时就让用户体验到个性化推荐服务这一问题。