# Week6-Thinking

### Thinking1 在推荐系统中，FM和MF哪个应用的更多，为什么

1.FM和MF的区别  
 MF 最早的鼻祖是SVD（奇异值分解）  
 1.MF 只考虑user和item特征，所以实际使用中有很多局限性；FM可以对更多维度进行预测，会将user和item都进行one-hot编码（one-hot读热编码就是把它变成[0,1]的数据构造）作为特征，使得特征维度非常巨大且稀疏。  
 2.FM会引入更多的辅助信息Side information作为特征，比如时间特征，上个电影看的是什么时间，而且特征和特征之间还有很多的特征关联，即引入了更多的特征维度  
 3.MF是FM的一个特例，即特征只有User ID和Item ID的FM模型。  
 4.使用场景不一样：MF只适用于评分预测，进行简单的特征计算，无法利用其他特征；FM可以计算二阶的特征，泛化能力比较强，即应用场景比较多。

2.FM在推荐系统，以及应用场景中的作用  
 主要可以解决以下问题  
 1.回归问题  
 损失函数就是预估值和实际值之间的最小二乘法，即误差的平方  
 2.二分类问题（CTR的预估问题）  
 FM的输出还可以通过sigmod函数变换，最后映射到一个[0,1]区间里，如果预估值y'<0 即它的sigmod函数就是小于0.5，y'> 0 即它的sigmod函数就是大于0.5，最终接近于1。

### Thinking2 FFM与FM有哪些区别？

1. FM 每个特征只有一个隐向量，FFM每个特征有多个隐向量  
 2. FM 是 FFM的特例，当FFM的k=1时就退化成FM  
 3. 隐向量的长度为 k，FFM的二次参数有 nfk 个，多于FM模型的 nk 个  
 4.由于隐向量与field相关，FFM二次项并不能够化简，计算复杂度比FM大，但是FFM的k值一般远小于FM的k值

5.FFM隐分类划分的更细，准确率更高

### Thinking3 DeepFM相比于FM解决了哪些问题，原理是怎样的

1.FM只考虑2阶特征组合，DeepFM既考虑低阶（1阶+2阶），又能考虑到高阶特征  
 2.DeepFM的特征组合本身就是对现实世界真实的一种模拟  
 3.采用end-to-end,不需要人工指定特征和特征之间怎么去组合，效果会更好，无需特征工程

4. DeepFM在各种benchmark和工程中效果好

DeepFM的原理  
包含5个部分  
1.Sparse Features 输入层，输入多个稀疏特征，所以需要做嵌入  
2. Dense Embeddings 对于输入层，做one-hot编码  
 对于每个稀疏特征做 embedding,学习到他们的embedding向量（维度相等，均为k）因为需要将这些embedding向量送到FM层做内积，同时embedding进行了降维，更好发挥了 Deep Layer的高阶学习能力

3.FM Layer  
 一阶特征：原始特征相加  
 二阶特征：原始特征embedding后的embedding向量两两内积  
4.Deep Layer  
 将每个embedding向量做级联，然后做多层的全连接，学习更深的特征  
5.Output Units  
 将FM层输出与Deep层输出进行级联，接一个dense层，作为最终输出结果  
  
end-to-end FM+DNN组合实际上是一个共享特征输入，共享权重  
中间的过程叫子网络，整个输入都是一样的，都是把输入文件从相同的地方输入，经过压缩的Dense Embedding（这个阶段，对于FM，输入到FM模型中可以获得特征的2阶表示，输入到DNN模型可以得到高阶特征表示）, 最后即给了FM，又给了DNN的部分，所以是一个共享权重的概念  
最后分别进行sigmoid,最后相加，===》y'=sigmoid(y(FM)+y(DNN))

FM部分：  
1.1阶特征建模  
2.通过隐向量点乘，得到2阶特征组合  
DNN部分：  
原始特征向量维度非常大，高度稀疏，为了更好的发挥DNN模型学习高阶特征能力，设计子网络（从输入层到嵌入层 Input Layer==>Embedding Layer）, 将原始的稀疏表示特征映射成为稠密的特征向量, 在filed中，因为filed是一个离散化的one-hot编码，所以整个filed中肯定只有一个位置是1，剩余都是0，所以不同的filed特征长度不同，但是子网络输出的向量具有相同的维度k,利用FM模型的隐特征向量V作为网络权重的初始化来获得子网络输出向量.

### Thinking4 Surprise工具中的baseline算法原理是怎样的？BaselineOnly和KNNBaseline有什么区别？

baseline算法：基于统计的基准预测打分，实际上有两个bias,用户之间的bias和商品的bias，实际上有一个优化算法在里面，可以使用SGD或者ALS  
 使用ALS进行优化  
 Step1，固定bu，优化bi  
 Step2，固定bi，优化bu  
  
 BaselineOnly 没有考虑协同过滤，在系统上面做了一个整体的偏好  
 KNNBaseline 在KNN上又加了一个baseline的模型，是两个模型的一个组合；考虑用户的偏差，偏差计算基于baseline

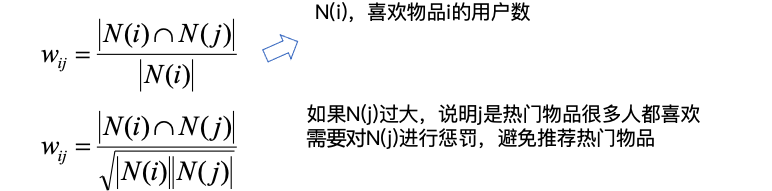
### Thinking5 基于邻域的协同过滤都有哪些算法，请简述原理

基于邻域的协同过滤(neighborhood-based)  
 UserCF：给用户推荐和他兴趣相似的其他用户喜欢的物品  
 ItemCF：给用户推荐和他之前喜欢的物品相似的物品

基于用户的协同过滤（UserCF）  
 利用行为的相似度计算用户的相似度  
 Step1，找到和目标用户兴趣相似的用户集合（相似度计算三种方式）  
 Step2，用户u对物品i的相似度，等价于K个邻居对物品i的兴趣度

Text

Description automatically generated  
 Step3，为用户u生成推荐列表  
 把和用户兴趣相同的k个邻居，喜欢的物品进行汇总，去掉用户u已经喜欢过的物品，剩下按照从大到小进行推荐  
  
  
基于物品的协同过滤（ItemCF）  
 利用行为的相似度计算物品的相似度  
 Step1，计算物品之间相似度

  
 Step2，用户u对物品i的兴趣度，等价于物品i的K个邻居物品，受到用户u的兴趣度

Text

Description automatically generated  
 Step3，为用户u生成推荐列表  
 和用户历史上感兴趣的物品越相似的物品，越有可能在用户的推荐列表中获得比较高的排名；预测用户u对物品的兴趣度，去掉用户u已经喜欢过的物品，剩下按照从大到小进行推荐

工具 Surprise工具中的基于邻域的协同过滤算法使用

https://surprise.readthedocs.io/en/stable/prediction\_algorithms\_package.html  
基于邻域的协同过滤算法  
knns.KNNBasic  
 k，邻域的个数 默认为40  
 sim\_options设置邻域参数，包括：  
 user\_based，是否为基于用户的协同过滤，默认为True，也可以设置为False（相当于item\_based）  
 name，相似度计算方式，默认为MSD，也可设置为cosine，pearson，pearson\_baseline  
 min\_support，最小支持度，对用户或者商品进行筛选  
 shrinkage：收缩参数（仅与Pearson correlation相似度相关）。 默认值为100  
knns.KNNWithMeans 就是减去一个均值  
 k，邻域的个数 默认为40  
 sim\_options设置邻域参数，包括：  
 user\_based，是否为基于用户的协同过滤，默认为True，也可以设置为False  
 name，相似度计算方式，默认为MSD，也可设置为cosine，pearson，pearson\_baseline  
 min\_support，最小支持度，对用户或者商品进行筛选  
 shrinkage：收缩参数（仅与Pearson correlation相似度相关）。 默认值为100  
knns.KNNWithZScore 减去一个均值，再除以一个标准差  
 k，邻域的个数 默认为40  
 sim\_options设置邻域参数，包括：  
 user\_based，是否为基于用户的协同过滤，默认为True，也可以设置为False  
 name，相似度计算方式，默认为MSD，也可设置为cosine，pearson，pearson\_baseline  
 min\_support，最小支持度，对用户或者商品进行筛选  
 shrinkage：收缩参数（仅与Pearson correlation相似度相关）。 默认值为100  
knns.KNNBaseline 考虑到用户打分时的偏差,用baseline替代了均值；时间慢，效果好  
 k，邻域的个数 默认为40  
 sim\_options设置邻域参数，包括：  
 user\_based，是否为基于用户的协同过滤，默认为True，也可以设置为False  
 name，相似度计算，推荐使用pearson\_baseline  
 bsl\_options，baseline优化方式，可以设置SGD,ALS等优化算法  
 min\_support，最小支持度，对用户或者商品进行筛选  
 shrinkage：收缩参数（仅与Pearson correlation相似度相关）。 默认值为100