**Thinking1:奇异值分解SVD的原理是怎样的，都有哪些应用场景?**

答: 1.原理:本质上是对矩阵的分解,但是与特征分解不同,SVD不要求分解的矩阵为方阵,

假设矩阵A为一个的矩阵,那么我们定义矩阵A的SVD为:,其中是

一个矩阵,是一个矩阵,,除了主对角线上的元素以外全为0,主对角线

上的每个元素都称为奇异值,是一个的矩阵.和都是酉矩阵,即满足

然后将A和A的转置做矩阵乘法,得到一个方阵,最终求出SVD.

2.可以用于推荐系统的矩阵分解,可以用于图片压缩,可以用于数据降维,也可以用于信息检索.

**Thinking2:funkSVD, BiasSVD，SVD++算法之间的区别是怎样的?**

答: 1.funkSVD: 需要设置k,避开稀疏问题,只使用两个矩阵进行相乘,但是这样噪音比较

严重

2.BasicSVD: 加入了用户偏好和商品偏好,将与个性化无关的部分,设置为偏好部分.

3.SVD++: 在BasicSVD算法上进行了改进,考虑用户的隐式反馈.

**Thinking3:矩阵分解算法在推荐系统中有哪些应用场景，存在哪些不足?**

答: 矩阵分解的核心是将一个非常稀疏的评分矩阵分解为两个矩阵,一个表示user的特性,

一个表示item的特性,将两个矩阵中各取一行和一列向量做内积就可以得到对应评分.

所以一般来说矩阵分解在推荐系统中常常用来做评分预测.

但矩阵分解也存在不足,因为只考虑user和item特征,对于其他特征的利用我们需要

使用新的工具

**Thinking4:** **item流行度在推荐系统中有怎样的应用?**

答: 1.冷启动中的使用

2.协同过滤中的TopN推荐

3.判断用户偏好,进行个性化推荐

**Thinking5:** **推荐系统的召回阶段都有哪些策略?**

答: 1.协同过滤:主要分为基于用户的协同过滤,基于物品的协同过滤和基于模型的协同过

滤(矩阵分解)

2.向量化召回: 主要通过模型来学习用户和物品的兴趣向量，并通过内积来计算用户和

物品之间的相似性,从而得到最终的候选集

3.深度树检索:通过构造一棵兴趣树,来进行推荐.

**Action1: 选择任意一张图片，对其进行灰度化，然后使用SVD进行图像的重构，当奇异值数量为原有的1%，10%，50%时，输出重构后的图像?**

****

**Action2: 使用Google Colab编辑器，对MovieLens数据集进行评分预测，计算RMSE（使用funkSVD, BiasSVD，SVD++）**

** **

**Action3: 使用PowerBI（或其他工具），对nCoV数据进行可视化呈现**