# Week5-Thinking

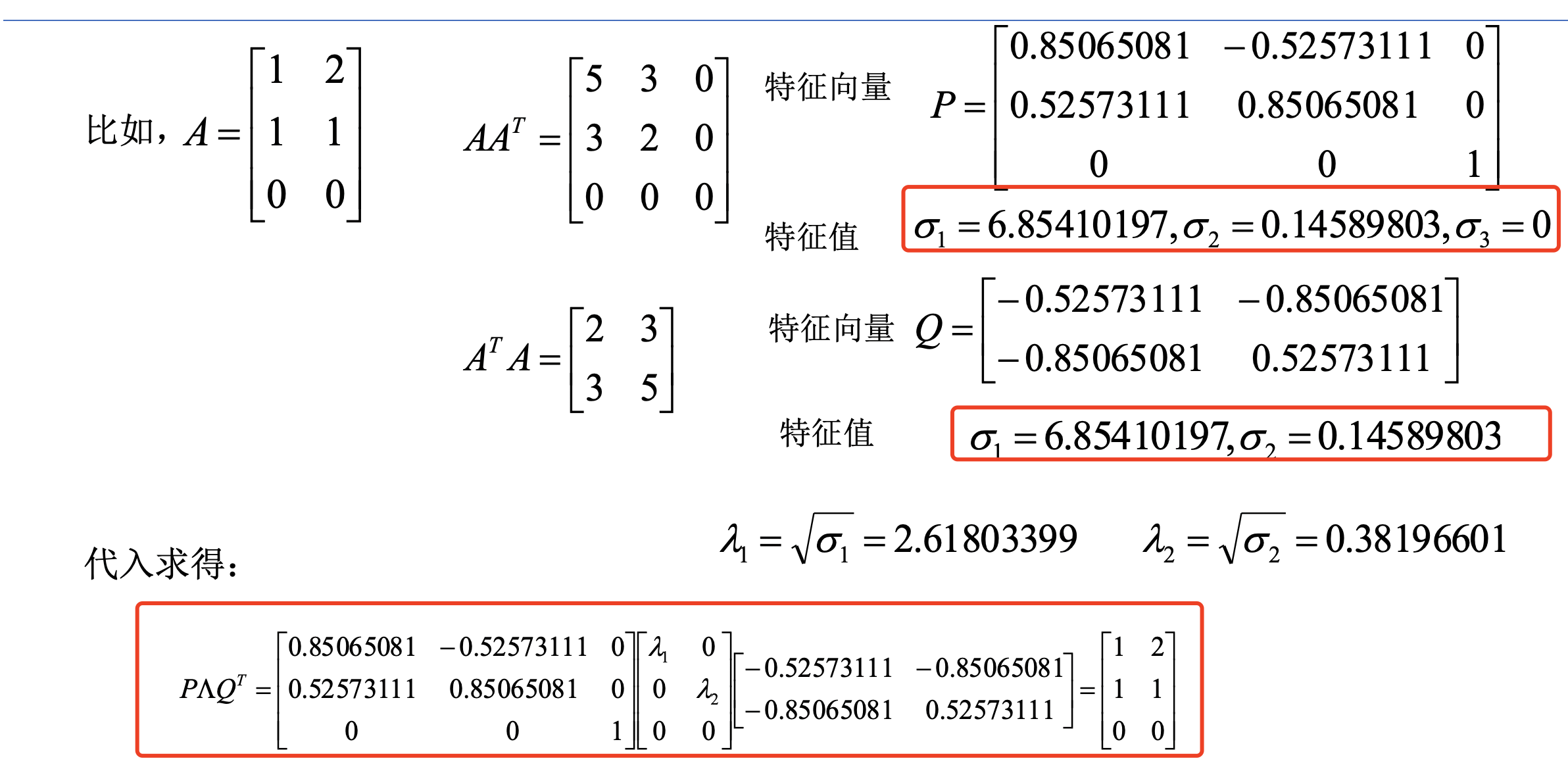
### Thinking1 奇异值分解SVD的原理是怎样的，都有哪些应用场景

1奇异值分解的原理

奇异值分解的降维处理主要体现在：只选取前k个最大的奇异值和其对应的特征向量

Graphical user interface, application

Description automatically generated



奇异值分解：



λ1为特征值，p1为左奇异矩阵的特征向量

q1为右奇异矩阵的特征向量

奇异值分解可以通过三个矩阵对原矩阵进行一个等价的还原，拆解成不同维度上特征的叠加

SVD最大的价值===》重构降维，就是帮助我们抽取出来特征值和特征向量，以至于我们对于k（特征值的个数）的值不需要取太大  
 举例k=50的时候就能够90%对图像进行还原，k=500和k=50的时候变化不大

svd的不足在于只考虑了交互信息，没考虑其他特征，以及矩阵分解是线性变换，无法捕捉非线性特征。

2、举例说明两个以上的使用场景  
 使用场景：

1.对图像进行压缩重构  
 2. 信息检索  
 3. 推荐系统应用-计算物品item或者用户user之间的相似度，可以用SVD将原始数据映射到低维空间中，然后节省计算相似度时的计算资源

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

1、能简述3种算法之间的差异（10points)  
funkSVD需要设置k，来对矩阵近似求解，矩阵分解之后的还原，只需要关注与原来矩阵中有值的位置进行对比即可，不需要对所有元素进行对比  
避开稀疏问题，而且只用两个矩阵进行相乘  
  
BiasSVD算法在funkSVD基础上加上了偏好部分，用户和商品本身之间有一个偏好  
用户有自己的偏好(Bias)，比如乐观的用户打分偏高  
商品也有自己的偏好(Bias)，比如质量好的商品，打分偏高  
将与个性化无关的部分，设置为偏好(Bias)部分  
  
SVD++算法在BiasSVD算法基础上进行了改进，考虑用户的隐式反馈  
隐式反馈：没有具体的评分，但可能有点击，浏览等行为  
在考虑用户隐式反馈的情况下，最终得到P和Q

### 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

1、在推荐系统中的典型应用场景  
 在推荐系统中，可以将user-item评分问题，转化为SVD矩阵分解  
2、MF在推荐系统中的局限性  
 1）矩阵往往是稀疏的，有大量的缺失值，SVD分解要求矩阵是稠密的，且计算量极大  
 2）填充方式简单粗暴，就会带来噪音非常大的问题

### Thinking4 假设一个小说网站，有N部小说，每部小说都有摘要描述。如何针对该网站制定基于内容的推荐系统，即用户看了某部小说后，推荐其他相关的小说。原理和步骤是怎样的

步骤：  
1.对小说摘要描述进行特征提取  
 N-Gram，提取N个连续字的集合，作为特征  
 IF-IDF, 按照（min\_df,max\_df）提取关键词，并生成IF-IDF矩阵  
2.计算小说之间的相似度矩阵-余弦相似度  
3.对于用户看过的某部小说，选择相似度最大的TopK部小说进行输出推荐  
  
基于内容的推荐系统：  
1.依赖性低，不需要动态的用户行为，只要有内容就可以进行推荐  
2.系统不同阶段都可以应用  
 1）系统冷启动，内容是任何系统天生的属性，可以从中挖掘到特征，实现推荐系统的冷启动。一个复杂的推荐系统是从基于内容的推荐成长起来的  
 2）商品冷启动，不论什么阶段，总会有新的物品加入，这时只要有内容信息，就可以帮它进行推荐

### Thinking5 Word2Vec的应用场景有哪些

NLP:  
文本处理，Word2Vec解决的是单词word和文章doc的问题，计算出单词和单词之间的相似度，embedding成一个固定的维度，向量和向量之间可以进行一个相似度的计算  
推荐系统：  
1.微博大v的推荐，关注了某个大V，微博就会推荐相似度极高的大V。  
 把大V类比成一个单词，将每一个用户关注大v的顺序类比成文章，这样就可以Embedding化，大V和大V之间就可以进行相似度的推荐。  
2.商品的推荐  
 一个商品item就可以看成一个单词，用户对商品行为的顺序（收藏，加购物车，点击）就可以看成一篇文章，所有item收藏加购物车的顺序就可以理解成一篇文章，  
 就可以预测，如果收藏了这些内容，这些收藏的内容和内容之间相似度比较高的商品有哪些。  
3.广告主在媒体网站上打广告，媒体网站让广告主自行决定要将广告推荐给哪些目标人群。  
 每一个页面就是一个词，将每个用户浏览的页面，按照浏览的顺序排列，形成文章。  
 这样，根据训练后的词向量，就可以计算出页面之间的相似程度，然欧进行推荐。