**假设所有数据均使用float64格式存储，占用8个字节**

**原始评分表的大小为，实际存储需要消耗约229.2GB**

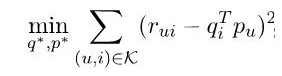
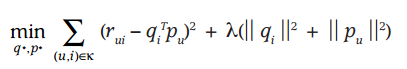
**SVD：**

特征值分解，，通过选取一部分特征值来最大程度上还原原始矩阵，并且起到压缩的效果，假设取个最大的特征值，数据的压缩比为。但SVD存在的明显的缺陷：1，在矩阵稀疏的情况下信息利用的效率非常低，并且不够准确 2，计算过程中需要分解、降维、求逆等运算，复杂度。

实际测试：完整计算大约需要核小时，没有完成完整计算，RMSE约0.796，压缩比约为1

**FunkSVD：**

使用两个低秩特征矩阵来描述用户与书籍的特点，相比SVD能够大幅压缩存储内容，并通过梯度下降寻找最优的矩阵。其中为了避免过度拟合，采用了带有L2正则的优化函数。

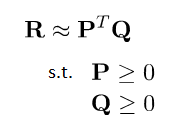
 

实际测试：RMSE= 1.247915818712198，运行时间核小时，数据压缩比为，

加载的准备时间更长，但是加载后计算速度加快，适用于处理大型数据。

**NMF：**

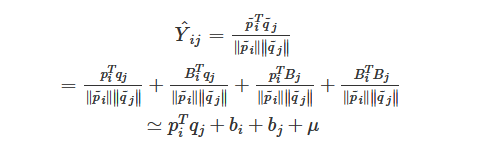
在低秩分解的基础上增加了非负约束，能够在实际数据中产生更好的效果。

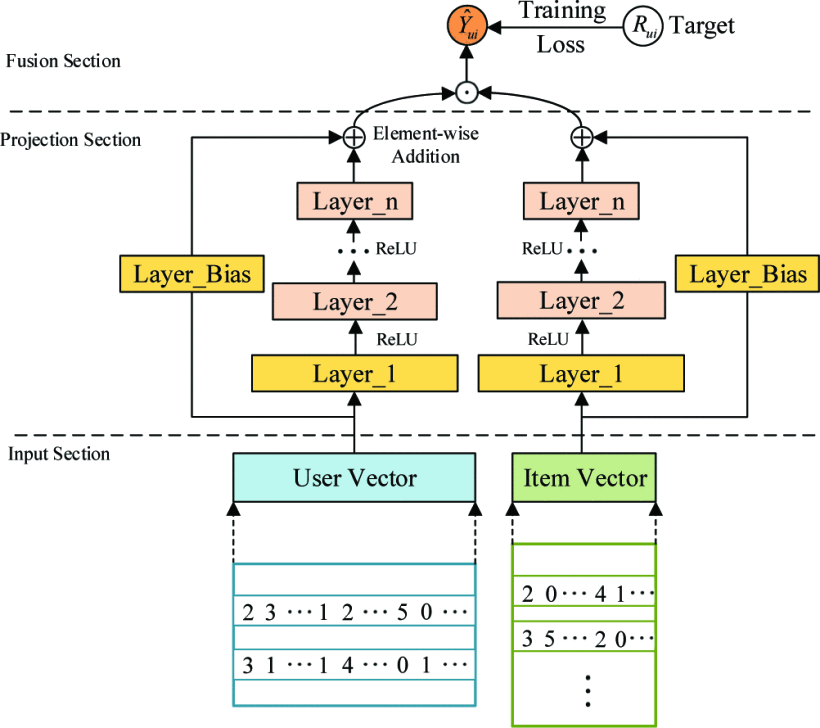


实际测试：与FunkSVD比较相似，采用50个隐藏维度，RMSE=0.79537，运行时间核小时，数据压缩比为

**BDMF：**

使用机器学习的方式为每个用户与书籍计算1维的特征向量，并通过两个向量的相似程度判断网络计算最后预测的评分。BDMF在DMF的基础上添加了包含个人偏好的Bias项进入最后的判断当中，用以保留原始的个人偏好信息。





实际测试：每个对象50个隐藏维度，训练10个epoch，数据压缩比为，RMSE=3.7529，RMSE偏高原因为对所有数据进行训练，而非矩阵分解对于小块数据处理。

机器学习的模式相比其他模式最大的优势在于能够使用列表格式的数据进行训练，而矩阵分解的模式只能对矩阵进行局部分块的处理，这些处理是复杂并且有局限性的，机器学习则能够极大程度地缓解内存的压力。