

# 推荐系统综述：

## 一 SVD：矩阵分解

其中关于 SVD 的理论及推导有兴趣可以查阅其他资料；这里不做描述。

提到了将用户-商品矩阵进行分解，从而实现对未打分项进行打分。矩阵分解是指将一个矩阵分解成两个或者多个矩阵的乘积。对于上述的用户-商品矩阵(评分矩阵)，记为  $R_{U \times I}$ ，可以将其分解成两个或者多个矩阵的乘积，假设分解成两个矩阵  $P_{U \times K}$  和  $Q_{K \times I}$ ，我们要使得矩阵  $P_{U \times K}$  和  $Q_{K \times I}$  的乘积能够还原原始的矩阵  $R_{U \times I}$ ：

$$R_{U \times I} \approx P_{U \times K} \times Q_{K \times I} = R^u \times R^i$$

其中，矩阵  $P_{U \times K}$  表示的是  $u$  个用户与  $k$  个主题之间的关系，而矩阵  $Q_{K \times I}$  表示的是  $k$  个主题与  $i$  个商品之间的关系。

SVD 应用于推荐系统时，其实只是用了其结论，即 user 和 item 的评分矩阵  $R$  可以写成两个矩阵的乘积：

$$R_{U \times I} = P_{U \times K} Q_{K \times I}$$

假设存在以下的 user 和 item 的评分矩阵  $R$ ：其中的值表示，user 对 item 的打分

User / item	1	2	3	4	5	.....
1	5	4	4.5	?	3.9	.....
2	?	4.5	?	4.5	?	.....
3	4.5	?	4.4	4	4	.....
4	?	4.8	?	?	4.5	.....
5	4	?	4.5	5	?	.....
.....	.....	.....	.....	.....	.....	.....

由于并不是所有的用户对所有的 item 均有评分记录，因此这里的“？”就表示了 user 对 item 没有记录的值，也就是我们在做推荐系统时需要预测的值，那么问题就是我们需要使用已有的评分记录去预测未知的评分记录，这里就有定义：

$$R_{U \times I} = P_{U \times K} Q_{K \times I}$$

这里的  $u$  表示用户数， $i$  表示商品数，因此可以利用  $R$  中已知的评分数据去训练  $P$  和  $Q$  矩阵，并且最小化一、矩阵分解：

SVD：其用户评分计算如下：其中  $b_u$  表示用户的 bias， $b_i$  表示 item 的 bias， $\mu$  表示一个基本的 baseline

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

通过最小化损失函数，来得到最终的推荐模型，其中采用 SGD 优化方法：

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$$

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u)$$

$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i)$$

SVD++：某个用户对某个电影进行了评分，那么说明他看过这部电影，那么这样的行为事实上蕴含了一定的信息，因此我们可以这样来理解问题：评分的行为从侧面反映了用户的喜好，可以将这样的反映通过隐式参数的形式体现在模型中，从而得到一个更为精细的模型，便是 SVD++。

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left( p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

其中  $I(u)$  为该用户所评价过的所有电影的集合， $y_j$  为隐藏的“评价了电影  $j$ ”反映出的个人喜好偏置。

NMF: Non-negative Matrix Factorization 非负矩阵分解:

相比于 SVD，这要求矩阵中所有的值为非负值（考虑评分的有效性），

$$\hat{r}_{ui} = q_i^T p_u,$$

由于需要保证参数在更新过程中的非负性，因此所有的参数都以如下的方式进行更新，并且这种更新方式等价于 SGD 中的加减法

$$p_{uf} \leftarrow p_{uf} \cdot \frac{\sum_{i \in I_u} q_{if} \cdot r_{ui}}{\sum_{i \in I_u} q_{if} \cdot \hat{r}_{ui} + \lambda_u |I_u| p_{uf}}$$

$$q_{if} \leftarrow q_{if} \cdot \frac{\sum_{u \in U_i} p_{uf} \cdot r_{ui}}{\sum_{u \in U_i} p_{uf} \cdot \hat{r}_{ui} + \lambda_i |U_i| q_{if}}$$

二、基于近邻（KNN）的推荐:

**KNNBasic:**

基于用户：最相似的  $k$  个用户对某个电影的评分的加权值；

基于物品：该用户对最相似的  $k$  个物品的评分的加权值；

如何衡量距离？：各种相似度、距离衡量；

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)}$$

**KNNWithMeans:** 由于用户打分存在偏好，如某些用户喜欢打高分，某些用户喜欢打低分，因此需要减去用户的打分偏差；

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - \mu_v)}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

$$\hat{r}_{ui} = \mu_i + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - \mu_j)}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)}$$

**KNNWithZScore:** 仍然是为了出去用户的打分偏差因素；只不过采用了 Z-score;

$$\hat{r}_{ui} = \mu_u + \sigma_u \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - \mu_v) / \sigma_v}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

$$\hat{r}_{ui} = \mu_i + \sigma_i \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - \mu_j) / \sigma_j}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)}$$

**KNNBaseline:** 考虑了基线评价；

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)}$$

**Co-clustering:** 基于聚类的推荐;

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i}),$$

where  $\overline{C_{ui}}$  is the average rating of co-cluster  $C_{ui}$ ,  $\overline{C_u}$  is the average rating of  $u$ 's cluster, and  $\overline{C_i}$  is the average rating of  $i$ 's cluster. If the user is unknown, the prediction is  $\hat{r}_{ui} = \mu_i$ . If the item is unknown, the prediction is  $\hat{r}_{ui} = \mu_u$ . If both the user and the item are unknown, the prediction is  $\hat{r}_{ui} = \mu$ .