

# 融合邻域模型与矩阵分解模型的推荐算法<sup>①</sup>

张 航, 叶东毅

(福州大学 数学与计算机科学学院, 福州 350108)

**摘 要:** 协同过滤推荐算法是目前构建推荐系统最为成功的算法之一, 它利用已知的一组用户对物品喜好数据来对推测用户对其他物品的喜好, 其中, 能够直接刻画用户与项目潜在特征的矩阵分解模型和通过分析物品或者项目间相似度的邻域模型是研究的热点. 针对这两个模型存在的不足, 提出了一种将邻域模型与矩阵分解模型有效结合的方法, 进而构建了一个改进的协同过滤推荐算法, 提高了预测准确性. 实验结果验证了改进算法的正确性与有效性.

**关键词:** 推荐系统; 协同过滤; 矩阵分解模型; 邻域模型

## Recommender Algorithm Incorporating Neighborhood Model with Matrix Factorization

ZHANG Hang, YE Dong-Yi

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China)

**Abstract:** Collaborative Filtering(CF) is one of the most successful approaches for building recommender system, it uses the known preferences of a group of users to make predictions of unknown preferences of other users. The matrix factorization models which can profile both users and items latent factors directly, and the neighborhood models which can analyze similarities between users and items are current research focuses. A method of merging both matrix factorization models and neighborhood models is proposed, which can make further accuracy improvements. The experiment results show that this method is correct and feasible.

**Key words:** recommender system; collaborative filtering; matrix factorization model; neighborhood model

随着信息技术的迅猛发展, 互联网上存在着巨大信息源, 人们面临着严重的信息过载问题(information over-load)<sup>[1]</sup>, 导致在查找自己所需信息时耗费大量时间. 个性化推荐系统<sup>[2]</sup>提供了一种有效的方式, 帮助用户缩减信息获取过程耗费的时间, 获取满意的信息.

协同过滤技术(Collaborative Filtering, CF)<sup>[3]</sup>是至今为止最为成功的个性化推荐技术之一. 它利用用户历史行为(如用户历史交易记录或者用户的评分)来进行个性化的推荐. 协同过滤推荐算法包括基于邻域的模型与潜在因子模型(latent factor model)等, 其中, 邻域模型主要刻画物品或者用户之间的相似度. 一个基于物品的邻域模型通过计算用户对相似物品的评分来预测用户对该物品的评分<sup>[4]</sup>, 在某种意义上, 是将用户

映射入项目空间, 将用户视为已评分项目的集合; 基于矩阵分解(Matrix Factorization, MF)的潜在因子模型<sup>[5]</sup>的基本思想是把高维向量空间模型中的信息映射到低维的潜在语义空间中. 降维处理使得物品与用户的潜在关系能够在同一潜在子空间中自然的显现出来, 揭示隐含在复杂数据背后的简单结构, 在预测的稳定性以及精确度上获得了广泛的认可<sup>[6,7]</sup>. 协同过滤推荐技术的应用领域非常广泛, 其中 Tapestry<sup>[8]</sup>是最早的推荐系统之一, 该系统记录了每个用户对他们阅读文章的观点, 并且这些观点可以被其他用户进行获取. GroupLens/Net Perceptions<sup>[9]</sup>、Ringo/Firefly<sup>[10]</sup>以及 MovieLens 都是较早期的著名的推荐系统. 协同过滤推荐算法最大的优点是对推荐对象没有特殊要求, 能

① 基金项目: 国家自然科学基金(61473089)

收稿时间: 2015-10-10; 收到修改稿时间: 2015-12-02 [doi:10.15888/j.cnki.csa.005185]

够处理非结构化的复杂对象,如书籍、文章、网页、音乐、电影以及百货等<sup>[11]</sup>.

虽然协同过滤推荐系统有很多优点,但是也存在一些问題.文献[12]中指出,基于邻域模型和潜在因子模型着重于评分数据的结构的不同层次,都并非是最优的选择.邻域模型对于检测局部关系十分有效,它着重于用户或者项目的几个重要的邻居节点,而忽略了其他大量的评分,从而使得该方法不能完全捕捉包含在全体评分中的一些信息.潜在因子模型长于评估包含大量评分的全局结构,但却忽略了相近用户或者项目小集合间的强关联关系,而这恰恰是邻域模型的优点.文献[13]介绍了结合两种模型的方法,但这是一种后处理矩阵分解结果的方法,而不是将邻域模型与潜在因子模型统一在一个模型中.文献[4]介绍了一种融合邻域模型与潜在因子模型的改进协同过滤模型,该方法将用户隐性反馈信息融合入邻域模型,并将该邻域模型中的参数结合到潜在因子模型中.

本文提出一种结合邻域模型与潜在因子模型的新方法,该方法将邻域模型中的  $k$  近邻信息与潜在因子模型中的特征向量结合起来,能够在矩阵分解的同时,保证相近用户或者项目小集合间的强关联关系不被忽略,从而提高模型的精确度.

## 1 邻域模型与矩阵分解模型

### 1.1 邻域模型中的 $k$ 近邻

基于邻域的算法是推荐系统中最基本的算法,分为两大类,一类是基于用户的协同过滤算法,另一类是基于项目的协同过滤算法.这两类算法的关键在于计算两个用户或者项目之间的相似度,主要是利用行为的相似度来计算.对于相似度的计算,Badrul Sarwar等在文献[14]中做了详细的研究,本文将采用余弦相似度来计算评分矩阵用户间与项目间的相似度,使用平均分补全稀疏评分矩阵.计算获得用户之间与项目之间的相似度后,确定每个用户或者项目的最佳邻居个数( $k$ 近邻),形成每个用户的邻居用户集( $C^U$ 保存所有用户与其邻居用户集中用户的相似度),以及每个项目的邻居项目集( $C^V$ 保存所有项目与其邻居项目集中项目的相似度).

### 1.2 概率矩阵分解模型

本文引入文献[15]的概率矩阵分解(PMF)模型作为基本的矩阵分解模型的框架.

对于  $N$  个用户与  $M$  个项目,假设评分矩阵为  $R \in \mathfrak{R}^{N \times M}$ ,其中元素  $r_{ij}$  表示用户  $u_i$  对项目  $v_j$  的评分.记  $U \in \mathfrak{R}^{f \times N}$  和  $V \in \mathfrak{R}^{f \times M}$  分别为分解得到的用户以及项目的  $f$ -维特征矩阵,其列向量  $U_i$  和  $V_j$  分别表示相对应的用户  $u_i$  与项目  $v_j$  的潜在特征向量.

PMF 模型假设可观测评分是由模型  $r_{ij} = u + U_i^T V_j$  和高斯观测噪声组成的,  $u$  为可观测评分的平均值,因此可观测评分的条件概率分布可以定义为:

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^N \prod_{j=1}^M [N(r_{ij}|u + U_i^T V_j, \sigma_R^2)]^{I_{ij}} \quad (1)$$

其中,  $N(x|\mu, \sigma^2)$  表示  $x$  服从均值为  $\mu$ , 方差为  $\sigma^2$  的高斯分布,  $I_{ij}$  是指示函数,如果用户  $\mu_i$  对项目  $v_j$  有评分,其值为 1, 否则为 0.

同样,假设  $U$  与  $V$  服从均值为 0 的球形高斯先验:

$$p(U|\sigma_U^2) = \prod_{i=1}^N N(U_i|0, \sigma_U^2 I),$$

$$p(V|\sigma_V^2) = \prod_{j=1}^M N(V_j|0, \sigma_V^2 I) \quad (2)$$

经过贝叶斯推理,可以得到  $U$  与  $V$  的联合后验概率分布:

$$p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) \propto p(R|U, V, \sigma_R^2) p(U|\sigma_U^2) p(V|\sigma_V^2) = \prod_{i=1}^N \prod_{j=1}^M [N(r_{ij}|u + U_i^T V_j, \sigma_R^2)]^{I_{ij}} \times \prod_{i=1}^N N(U_i|0, \sigma_U^2 I) \times \prod_{j=1}^M N(V_j|0, \sigma_V^2 I) \quad (3)$$

对  $U$  与  $V$  的后验分布取对数,可得:

$$\ln p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) = -\frac{1}{2\sigma_R^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} [r_{ij} - (u + U_i^T V_j)]^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^N U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^M V_j^T V_j + C \quad (4)$$

其中,  $C$  是与超参数不相关的常数.

最大化对数的后验分布等价于最小化如下带正则项的误差平方和函数:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} [r_{ij} - (u + U_i^T V_j)]^2 + \frac{\lambda_U}{2} \sum_{i=1}^N \|U_i\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^M \|V_j\|_{Fro}^2 \quad (5)$$

其中,  $\lambda_U = \sigma_R^2 / \sigma_U^2$ ,  $\lambda_V = \sigma_R^2 / \sigma_V^2$ ,  $\|\bullet\|_{Fro}^2$  表示 Frobenius 范数。

## 2 基于模型融合的推荐算法

### 2.1 融合邻域模型与矩阵分解模型的改进模型

基于用户的协同过滤算法, 通过参考与目标用户具有相似兴趣和需求的其他用户的选择来决定如何为该用户进行信息推荐。其主要思想是: 如果两个用户对项目中的一些项目的评分比较相似, 则他们对其他项目的评分也比较相似<sup>[11]</sup>。因而他们的用户特征向量也是相似的。同理, 基于项目的协同过滤算法, 若两个项目相似, 则它们的项目特征向量也相似。

对于项目或者用户特征向量的相似度计算为:

$$\text{项目相似度: } \cos(V_j, V_t) = \frac{V_j^T \cdot V_t}{\|V_j\| \|V_t\|}$$

$$\text{用户相似度: } \cos(U_i, U_k) = \frac{U_i^T \cdot U_k}{\|U_i\| \|U_k\|}$$

因此, 假设用户或项目间相似度的条件概率分布为:

$$p(C^U | U, \sigma_{C^U}^2) = \prod_{i=1}^N \prod_{k=1}^M [N(C_{ik}^U | \frac{U_i^T \cdot U_k}{\|U_i\| \|U_k\|}, \sigma_{C^U}^2)]^{I_{ik}}$$

$$p(C^V | V, \sigma_{C^V}^2) = \prod_{j=1}^M \prod_{t=1}^M [N(C_{jt}^V | \frac{V_j^T \cdot V_t}{\|V_j\| \|V_t\|}, \sigma_{C^V}^2)]^{I_{jt}} \quad (6)$$

其中,  $C^U$ 、 $C^V$  为章节 1.1 中得到的 k 近邻的相似度集合,  $I_{ik}$  为指示函数, 在  $C^U$  中, 如果用户  $\mu_i$  与  $\mu_k$  有相似度, 其值为 1, 否则为 0;  $I_{jt}$  同理。

经过贝叶斯推理, 可以得到  $U$  与  $V$  的联合后验概率分布:

$$p(U, V | R, C^U, C^V, \sigma_R^2, \sigma_{C^U}^2, \sigma_{C^V}^2, \sigma_U^2, \sigma_V^2) \propto p(R | U, V, \sigma_R^2) p(C^U | U, \sigma_{C^U}^2) p(C^V | V, \sigma_{C^V}^2) p(U | \sigma_U^2) p(V | \sigma_V^2)$$

$$= \prod_{i=1}^N \prod_{j=1}^M [N(r_{ij} | u + U_i^T V_j, \sigma_R^2)]^{I_{ij}} \times \prod_{i=1}^N \prod_{k=1}^M [N(C_{ik}^U | \frac{U_i^T \cdot U_k}{\|U_i\| \|U_k\|}, \sigma_{C^U}^2)]^{I_{ik}} \times \prod_{j=1}^M \prod_{t=1}^M [N(C_{jt}^V | \frac{V_j^T \cdot V_t}{\|V_j\| \|V_t\|}, \sigma_{C^V}^2)]^{I_{jt}} \times \prod_{i=1}^N N(U_i | 0, \sigma_U^2) \times \prod_{j=1}^M N(V_j | 0, \sigma_V^2) \quad (7)$$

对  $U$  与  $V$  的后验分布取对数, 可得:

$$\ln p(U, V | R, C^U, C^V, \sigma_R^2, \sigma_{C^U}^2, \sigma_{C^V}^2, \sigma_U^2, \sigma_V^2) = -\frac{1}{2\sigma_R^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} [r_{ij} - (u + U_i^T V_j)]^2 - \frac{1}{2\sigma_{C^U}^2} \sum_{i=1}^N \sum_{k=1}^M I_{ik} [C_{ik}^U - \frac{U_i^T \cdot U_k}{\|U_i\| \|U_k\|}]^2 - \frac{1}{2\sigma_{C^V}^2} \sum_{i=1}^N \sum_{k=1}^M I_{jt} [C_{jt}^V - \frac{V_j^T \cdot V_t}{\|V_j\| \|V_t\|}]^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^N U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^M V_j^T V_j + C \quad (8)$$

最大化对数的后验分布等价于最小化如下带正则项的误差平方和函数:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} [r_{ij} - (u + U_i^T V_j)]^2 + \frac{\lambda_{C^U}}{2} \sum_{i=1}^N \sum_{k=1}^M I_{ik} [C_{ik}^U - \frac{U_i^T \cdot U_k}{\|U_i\| \|U_k\|}]^2 + \frac{\lambda_{C^V}}{2} \sum_{i=1}^N \sum_{k=1}^M I_{jt} [C_{jt}^V - \frac{V_j^T \cdot V_t}{\|V_j\| \|V_t\|}]^2 + \frac{\lambda_U}{2} \sum_{i=1}^N \|U_i\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^M \|V_j\|_{Fro}^2 \quad (9)$$

其中,  $\lambda_{C^U} = \sigma_R^2 / \sigma_{C^U}^2$ ,  $\lambda_{C^V} = \sigma_R^2 / \sigma_{C^V}^2$ ,  $\lambda_U = \sigma_R^2 / \sigma_U^2$ ,  $\lambda_V = \sigma_R^2 / \sigma_V^2$ ,  $\|\bullet\|_{Fro}^2$  表示 Frobenius 范数。

再对  $U$  与  $V$  中的每名用户与每个项目使用梯度下降的方法求解, 使  $E$  达到最小值:

$$\frac{\partial E}{\partial U_i} = \sum_{j=1}^M I_{ij} [(u + U_i^T V_j) - r_{ij}] V_j + \lambda_{C^U} \sum_{k=1}^M I_{ik} (C_{ik}^U - \frac{U_i^T \cdot U_k}{\|U_i\| \|U_k\|}) \frac{U_k}{\|U_i\| \|U_k\|} + \lambda_U U_i$$

$$\frac{\partial E}{\partial V_j} = \sum_{i=1}^N I_{ij} [(u + U_i^T V_j) - r_{ij}] U_i + \lambda_{C^V} \sum_{t=1}^M I_{jt} (C_{jt}^V - \frac{V_j^T \cdot V_t}{\|V_j\| \|V_t\|}) \frac{V_t}{\|V_j\| \|V_t\|} + \lambda_V V_j \quad (10)$$

这里, 将这种结合用户与项目相似度的方法称为 User and Item based Probabilistic Matrix Factorization (UIPMF); 同时, 上述方法中, 若只结合用户相似度, 称为 User based Probabilistic Matrix Factorization (UserPMF), 只结合项目相似度的称为 Item based Probabilistic Matrix Factorization (ItemPMF)。

### 2.2 算法的复杂度分析

假设有  $N$  个用户,  $M$  个项目, 用户近邻表与项目近邻表的离线计算时间复杂度为  $o(N \times N)$  与  $o(M \times M)$ ; 需要维护两张离线的相似度表, 若近邻数为  $k$ , 则空间复杂度为  $o(k \times N)$ ,  $o(k \times M)$ 。梯度下降的时间复杂度为  $o(pf(\sum_{i=1}^N \sum_{j=1}^M I_{ij} + \sum_{i=1}^N \sum_{k=1}^M I_{ik} + \sum_{j=1}^M \sum_{t=1}^M I_{jt}))$ , 其中  $p$  为迭代次数, 是评分集与相似度表的线性关系, 空间复杂度为  $o(f \times (N + M))$ 。

## 3 实验设计及结果分析

### 3.1 数据集及评价标准

为了测试算法的有效性, 本文采用了推荐系统常用的两个数据集: Netflix 数据集和 MovieLens 数据集。

Netflix 数据集是 Netflix Prize 比赛中使用的标准测试数据集, 本文从中随机抽取了 8662 名用户对 3000 部电影约  $3 \times 10^5$  条评分信息(评分密度约为 1.1%), 称为 toy\_Netflix 数据集. MovieLens 数据集是由 GroupLens 项目组提供, 本文采用的 MovieLens\_1M 数据集, 包含有 6040 名用户对 3952 部电影的约  $10^6$  条评分信息(评分密度约为 4.2%).

文献[16]对推荐系统领域内的各种不同的评价标准做了总结, 本文将采用检验推荐算法最常用的 RMSE(均方根误差)作为评价依据, 均方根误差越小, 表示算法性能越好.

$$RMSE = \sqrt{|S_{test}|^{-1} \sum_{(i,j) \in S_{test}} (\hat{r}_{ij} - r_{ij})^2} \quad (11)$$

其中,  $S_{test}$  是测试集合,  $|S_{test}|$  是  $S_{test}$  中的元素个数.

### 3.2 实验设计及结果

本文在这两个数据集上, 以 RMSE 为评价标准检验算法准确度, 设计了 2 组实验, 从不同的方面对算法的性能进行测试, 并采用 10-fold 交叉验证划分训练集与测试集. 实验中, 用户与项目潜在特征矩阵的维度  $f$  设为 10, 最大迭代次数设为 100 次; 同时, 对比算法 PMF 的  $\lambda$  参数设置为 0.02.

A 组实验在 toy\_Netflix 数据集与 MovieLens\_1M 数据集上, 测试算法在不同  $k$  近邻数下的表现. 实验中, 数据集的 90% 作为训练集, 同时, 我们将  $k$  近邻数分别设置为 5、10、20、40、80 和 160, 比较算法效果, 结果如图 1、图 2 所示. 在两个数据集上, UIPMF、UserPMF、ItemPMF 方法的预测误差都比 PMF 方法好, 其中, 结合了用户与项目相似度的 UIPMF 方法, 相比于只结合用户或者项目相似度的方法, 可以取得更好的结果, 说明结合相似度的方法, 能够在矩阵分解挖掘全局关系的同时, 保留用户与项目的强关联关系, 提高模型的精确度. 比较 UIPMF 方法在不同  $k$  近邻下的表现, 随着  $k$  近邻数的增加, 预测精确度越好, 但是当到达一定阈值后, 增长会减缓, 甚至精确度会下降, 这也正好反映强关联关系对于推荐精确度产生的影响, 选择合适的  $k$  值对获得最高精确度非常重要. 表 1、表 2 是算法在不同  $k$  值下的时间比较, 与 PMF 算法比较, 改进算法迭代过程增加的运算时间是  $K$  值的线性关系; 改进运算时间的增加, 主要在于用户近邻表与项目近邻表的计算, 本文采用文献[14]介绍的基于邻域的经典方法计算近邻表, 该方法已在各类实际推荐系统中

应用, 因此, 改进算法在实际应用中具有可行性.

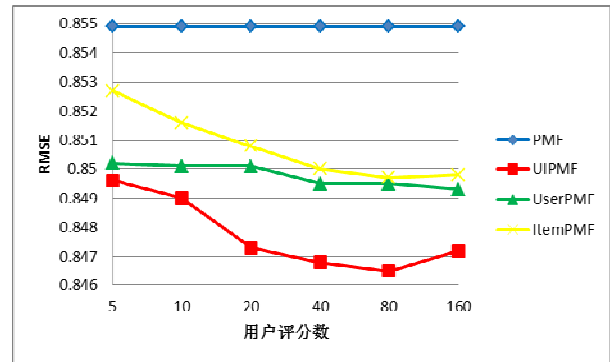


图 1 MovieLens 数据集下不同 k 近邻数比较

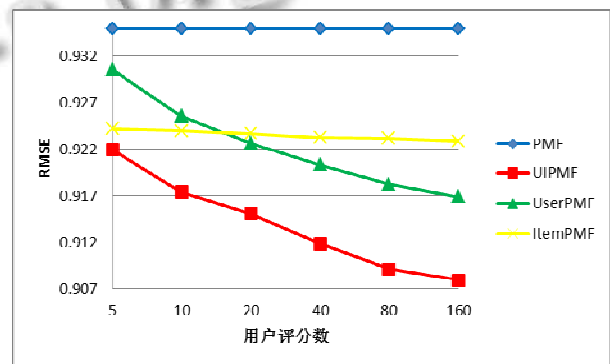


图 2 toy\_Netflix 数据集下不同 k 近邻数比较

表 1 MovieLens 数据集下不同 k 近邻数时间比较(s)

k 近邻	用户近邻表	项目近邻表	PMF	UIPMF	UserPMF	ItemPMF
5	82.6	55.7	190.8	195.4	193.1	192.5
10	83.6	56.0	190.8	202.0	197.2	195.5
20	86.6	57.4	190.8	208.5	202.2	199.0
40	91.2	59.2	190.8	225.6	213.4	205.6
80	97.8	62.1	190.8	261.1	234.3	218.1
160	115.8	70.1	190.8	334.4	276.5	248.4

表 2 toy\_Netflix 数据集下不同 k 近邻数时间比较(s)

k 近邻	用户近邻表	项目近邻表	PMF	UIPMF	UserPMF	ItemPMF
5	123.5	41.9	49.0	54.3	53.0	50.0
10	126.0	42.1	49.0	60.0	56.7	51.4
20	130.2	43.0	49.0	68.8	63.1	53.8
40	138.6	43.9	49.0	88.7	79.6	58.7
80	155.3	45.9	49.0	130.2	110.0	69.5
160	170.9	50.0	49.0	213.5	170.9	91.1

B 组实验在 toy\_Netflix 数据集与 MovieLens\_1M 数据集上, 测试不同稀疏度下算法的表现. 我们分别

从实验数据集中随机抽取 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20% 的数据作为训练集, 在不同的稀疏度下, 比较算法的效果. 实验结果如图 3、图 4 所示. UIPMF 在不同稀疏度下, 都能取得更好的结果, 特别是在 20%、30% 等比例的训练集中, UIPMF 的优势越明显: MovieLens\_1M 数据集上, 20%、30% 训练比例下, UIPMF 的 RMSE 比 PMF 降低约 5.1%、2.8%, toy\_Netflix 数据集上, 降低约 9.8%、7.8%. 我们发现, MovieLens\_1M 数据集的评分密度为 4.2%, 而 toy\_Netflix 数据集的评分密度为 1.1%, 而随着训练比例的下降, 评分密度会越来越小, 表明在评分密度稀疏时, UIPMF 比 PMF 的评分预测误差小. 实际应用中, 评分密度都在 1% 以下, 稀疏条件下的实验更能反映算法提取潜在信息的能力. 这说明结合邻域信息的 UIPMF 可以提取更多潜在信息的能力, 有效缓解数据稀疏性问题.

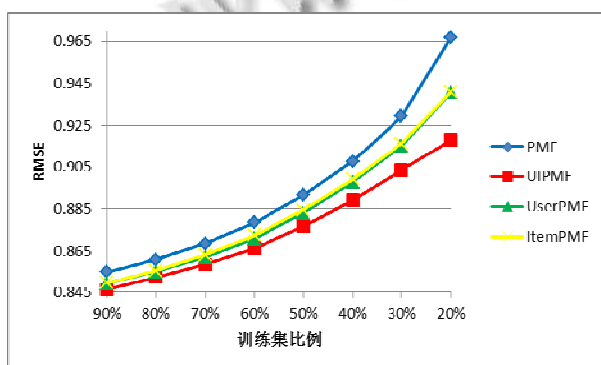


图 3 MovieLens 数据集下不同稀疏度比较

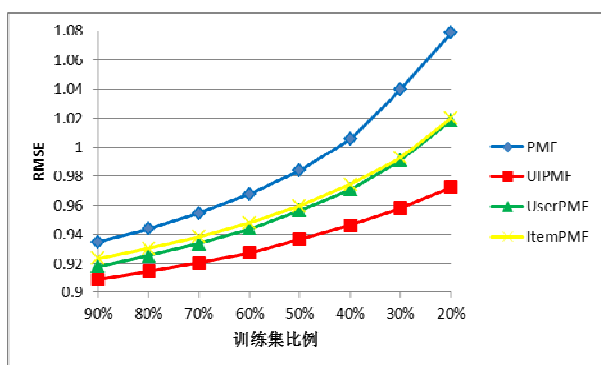


图 4 toy\_Netflix 数据集下不同稀疏度比较

#### 4 结论

本文针对邻域模型与潜在因子模型协同过滤算法推荐算法的不足, 提出一种融合两种模型的推荐算法, 在矩阵分解算法中融入了邻域信息, 综合考虑了全局

以及局部的信息. 在两种真实数据集上的实验结果表明, 该算法能够提高预测的准确度, 在稀疏条件下能更好地挖掘潜在信息, 有效缓解数据稀疏性问题.

在计算  $k$  近邻时, 本文采用了简单的余弦相似度, 在离线计算评分矩阵两用户或者项目之间的相似度时, 需要大量时间; 同时, 在计算相似度时, 没有充分利用数据集提供的用户与项目的属性信息, 在未来的工作中, 需要对这些方面加以改进.

#### 参考文献

- 1 项亮. 推荐系统实践. 北京: 人民邮电出版社, 2012: 39-44.
- 2 Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. Proc. of the 10th International Conference on World Wide Web. 2001. 285-295.
- 3 余力, 刘鲁. 电子商务个性化推荐研究. 计算机集成制造系统, 2004, 10(10): 1306-1313.
- 4 鲁权, 王如龙, 张锦, 等. 融合邻域模型与隐语义模型的推荐算法. 计算机工程与应用, 2013: 1303-0357.
- 5 Bell R, Koren Y, Volinsky C. Modeling relationships at multiple scales to improve accuracy of large recommender systems. Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07). New York, USA. 2007. 95-104.
- 6 Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. IEEE Computer, 2009, 42(8): 30-37.
- 7 Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature, 1999, 401(6755): 788-791.
- 8 Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry. Communications of the ACM. 1992.
- 9 Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews. Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work. 1994. 175-186.
- 10 Shardanand U, Maes P. Social information filtering: Algorithms for automating "Word of Mouth". Proc. of ACM CHI'95 Conference on Human Factors in Computing Systems, volume 1 of Papers: Using the Information of

- Others. 1995. 210-217.
- 11 姚平平,邹东升,牛宝君.基于用户偏好和项目属性的协同过滤推荐算法.计算机系统应用,2015,24(7):15-21.
- 12 Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proc of Acm Sigkdd Conference. 2008. 426-434.
- 13 Bell R, Koren Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. IEEE International Conference on Data Mining (ICDM'07). 2007. 43-52.
- 14 Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. Proc. of the 10th International Conference on World Wide WebACM. 2001. 285-295.
- 15 Salakhutdinov R, Mnih A. Probabilistic matrix factorization. Advances in Neural Information Processing Systems, 2008, 20(1): 1257-1264.
- 16 朱郁筱,吕琳媛.推荐系统评价指标综述.电子科技大学学报,2012,2:163-175.

[www.c-s-a.org.cn](http://www.c-s-a.org.cn)

[www.c-s-a.org.cn](http://www.c-s-a.org.cn)