

基于协同过滤和隐语义模型的混合推荐算法

徐吉¹, 李小波², 陈华辉¹, 许浩²

(1. 宁波大学 信息科学与工程学院, 浙江 宁波 315211;

2. 丽水学院 工学院, 浙江 丽水 323000)

摘要:协同过滤算法一般根据用户的评价信息来推测用户的喜好,但受到数据稀疏问题的影响,很多时候无法得到较为理想的推荐结果;除此之外,一般协同推荐算法忽略了用户兴趣的动态变化;文中提出的算法主要融合了相似度传递、用户兴趣迁移、隐语义模型等用以解决上述问题。首先提出了基于项目相似度的协同推荐算法。该算法深入研究了改进的余弦相似度方法,在执行过程中首先需要对项目进行信任关系建模,基于此来传递相似度,然后将这两部分相似度关系进行加权得到新的项目相似关系,可以将其应用到项目的评分中。其次,提出了基于用户兴趣迁移的隐语义模型推荐算法。该算法引入时间函数,重构用户的兴趣模型,实现对传统模型的修正,然后再使用梯度下降法来求解。最后,采用线性融合的办法,将以上两种算法进行融合。实验对比结果表明,混合推荐算法的推荐准确率较原先的算法有了较大的提高,因为它可以对丢失的信息进行补充,对于用户兴趣的变化能够较好的适应,同时大大弱化了数据的稀疏导致的一系列负面影响。

关键词:协同过滤;用户兴趣迁移;相似度传递;隐语义模型;混合推荐算法

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2020)02-0052-06

doi:10.3969/j.issn.1673-629X.2020.02.011

A Hybrid Recommendation Algorithm Based on Collaborative Filtering and Latent Factor Model

XU Ji¹, LI Xiao-bo², CHEN Hua-hui¹, XU Hao²

(1. School of Information Science and Engineering, Ningbo University, Ningbo 315211, China;

2. School of Engineering, Lishui College, Lishui 323000, China)

Abstract: The collaborative filtering algorithm generally estimates the user's preferences based on their evaluation information. However, due to the data sparsity, in many cases, it is impossible to obtain the ideal recommendation results. In addition, the collaborative recommendation algorithm ignores the dynamic change of user interest. The algorithm proposed in this paper mainly combines similarity transfer, user interest migration and latent factor model to solve the above problems. Firstly, we propose a collaborative recommendation algorithm based on project similarity transfer, which improves the cosine similarity method. In this algorithm, we model the trust relationship of the project and transfer the similarity based on it, then we weigh the similarity relationship between the two parts to get a new project similarity relationship, which can be applied to the project scoring. Secondly, we propose a latent semantic model recommendation algorithm based on user interest migration. The algorithm introduces the time function, reconstructs the user's interest model to implement the modification of the traditional model, and then uses the gradient descent method to solve. Finally, we use the linear fusion method to fuse the above two improved algorithms. The experiment shows that the linear fusion method is better because it can supplement the missing information, adapt to the change of user interest, and greatly weaken a series of negative effects caused by sparse data.

Key words: collaborative filtering; user interest migration; similarity transfer; latent factor model; hybrid recommendation algorithm

收稿日期:2019-03-20

修回日期:2019-07-23

网络出版时间:2019-11-07

基金项目:国家自然科学基金(61373057, 61572266)

作者简介:徐吉(1993-),男,硕士研究生,研究方向为大数据、数据挖掘、机器学习;李小波,博士,教授,CCF会员(43131M),研究方向为大数据、数据挖掘、生物信息学;陈华辉,博士,教授,研究方向为大数据处理、流数据处理、数据库、数据挖掘。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20191107.0912.056.html>

0 引言

随着互联网的发展,人们获取信息的方式愈加丰富,海量信息在满足需求的同时,也为人们带来一些困扰。大量信息中的无效信息一方面干扰了人们对正常信息的判断,另一方面也降低了人们对信息的处理效率^[1]。协同过滤算法^[2]一般根据用户的评价信息来推测用户的喜好,但受到数据稀疏问题^[2-3]的影响,很多时候无法得到较为理想的推荐结果;除此之外,一般协同推荐算法忽略了用户兴趣的动态变化;文中对传统协同过滤算法存在的上述问题进行了研究,并提出了改进后的协同过滤混合推荐推荐算法,用以解决上述问题。

就目前的现状来看,用的最多的当属协同过滤算法,它的突出缺点就在于解决数据稀疏性上表现不佳,导致在数据稀疏时,推荐性能大打折扣。LFM 算法^[3]是一种协同滤波算法,但是建立在模型的基础之上,而经过改进的 LFM 在缓和稀疏性问题上有一定的作用,但是它进行降维处理时容易造成数据的丢失^[4]。传统的推荐算法已经无法再产生较为精准的模型数据,究其原因,很大一部分是由于算法的局限性,当然,也不排除和数据本身的差异性有关。当前越来越多的推荐系统会选择融合多种策略,将各种算法的优点整合到一起,以克服单一算法的局限。

1 协同过滤算法

1.1 相似度计算方法

对于协同过滤算法来说,其最关键的内容是对用户或者是项目间的相似度进行计算^[5]。在这个过程中可以采用不同的计算方法,下面对其中几种比较典型的方法进行介绍。

(1) Pearson 相关系数法。

当前在推荐算法中已经较多地使用了 Pearson 相关系数法^[6-7]对用户或者项目间的相似关系进行计算,具体如下所示:

$$\text{sim}(u,v)=\frac{\sum_{i\in I}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I}(r_{vi}-\bar{r}_v)^2}}$$

(1)

其中, I 表示用户 u,v 都评分的项目集, r_{ui} 与 \bar{r}_u 分别表示用户 u 对项目 i 以及对全部项目的平均评分。

(2) 余弦相似度法^[8-10]。

假定两个用户 u,v 的项目评分向量表示为 \vec{u}, \vec{v} , 可以将二者的相似度表示为 $\text{sim}(u,v)$, 其公式如下所示:

$$\text{sim}(u,v)=\frac{\sum_{i\in I}r_{ui}-r_{vi}}{\sqrt{\sum_{i\in I}r_{ui}^2}\sqrt{\sum_{i\in I}r_{vi}^2}}$$

(2)

其中, r_{ui} 表示用户 u 对项目 i 的评分, I 表示用户 u,v 共同评分的项目集, \vec{u} 表示用户 u 对项目的评分向量。

在实际中不同的用户间往往存在明显的差异性,各个用户的评价标准也不相同。改进的余弦相似度^[9]公式如下所示:

$$\text{sim}(u,v)=\frac{\sum_{i\in I}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I}(r_{vi}-\bar{r}_v)^2}}$$

(3)

其中, \bar{r}_u 表示用户 u 对全部项目评分的平均值。

(3) Jaccard 相关系数法^[8,10]。

在很多情况下主体的特征属性值并不是连续的,在这些情形下无法直接对其相似度进行衡量,而是需要采用一些特殊的符号进行描述。有学者提出了 Jaccard 相关系数法,其公式如下所示:

$$\text{Jaccard}(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

(4)

对于此类问题可以采用 1/-1 代表用户喜欢/不喜欢,0 代表未标注,可以将用户 U, V 间的 Jaccard 相关系数计算公式表示为:

$$\text{Jaccard}(U,V)=\frac{\sum_{i=1}^m1\{u_i=v_i\wedge u_i\neq 0\}}{\sum_{i=1}^m1\{u_i\neq 0\vee v_i\neq 0\}}$$

(5)

其中, u_i 表示用户 u 对项目 i 形成的标注, $1\{*\}$ 表示指示函数。

1.2 改进的项目相似度计算方法

相似度计算结果的准确性比较依赖于数据量的大小,如果数据量较小,往往难以得到较好的效果。这些方法也比较容易受到数据稀疏性的影响,这必然会造成相似度计算结果的不准确。可以采用合理的方式对其相似度计算方法进行优化,如果两个项目间没有共同评分项目,此时无法直接对两个主体间的相似度进行计算,其中的稀疏评分矩阵即如表 1 中所示。

表 1 稀疏评分矩阵

	U_1	U_2	U_3	U_4	U_5
a	2	5			
b			5	4	4
c	5	5		5	3
d		4	4	3	
e	3	3	4		5
f	1	4	2	4	2

通过表中数据可以明显地看到,项目 a, b 不存在

共同评分用户,因此难以直接对项目 a, b 的相似度进行计算。经过分析可以发现出现这种问题的主要原因是数据的稀疏性。在表 1 中虽然没有直接对项目 a, b 共同评分的用户,但可以发现其他项目与 a 之间含有一些共同评分用户,其中的一些项目与 b 也有明显的相似度关系。此时可以根据项目 a 与其余项目间的相似度关系构建可信模型,然后可以间接得到项目 a 与项目 b 间的相似度关系。在表中可以发现 c, d, e, f 和 a 都含有可信关系,并且与 b 也有一定的相似性,因此可以通过间接的方式得到二者的相似度,具体的传递方式是 $b \rightarrow f \rightarrow a, b \rightarrow e \rightarrow a$ 。

1.3 可信关系建模

首先是根据项目来进行可信关系建模,然后将其应用到相似度的传递过程中。在日常生活中当有人向你推荐他信任的物品时,你会有较高的可能性去接受这种推荐,因此人们之间存在一定的信任关系。在对协同过滤算法进行设计时可以考虑这种信任关系,然后将其应用到信任网络的构建中。然后可以进行可信关系建模并完成对相似度的传递。基于上述方式能够实现间接相似度的计算。

由于大量的用户可能会对这两个项目进行评分,各个用户的评分存在一定的差异性,如果这种差异性不明显,这说明推荐的效果比较好。一般可以将用户对项目的评分情况设置一个阈值,当用户的评分在阈值范围内时,即认为项目之间存在较为可靠的信任关系。

准确推荐的含义可以表述为:用户 a, b 都对项目 i 进行评分,并且其评分结果在阈值 θ 之内。准确度可以表示为如下公式:

$$\text{Success}(a, b, i) \Leftrightarrow |R_{a,i} - R_{b,i}| \leq \theta \quad (6)$$

根据相同的定义,当用户 u 对两个项目 i, j 都评分,而用户 a 对项目 i 进行评分,此时可以将正确推荐的公式表示为:

$$\text{Success}(i, j, u) \Leftrightarrow |R_{u,i} - R_{u,j}| \leq \theta \quad (7)$$

项目 j 可以正确推荐项目 i 的集合如下所示:

$$\text{SuccessSet}(i, j) = \{(u, v, i) : \text{success}(u, v, i)\} \quad (8)$$

项目间的信任关系可以表示为:

$$\text{rate}(i, j) = \frac{|\text{success}(i, j, u)|}{|\text{item}(i) \cup \text{item}(j)|} \quad (9)$$

基于先前的分析可知,正确推荐所占据的比例对存在明显的影响。此外,信任关系还与其自身可靠性相关。但是还需要注意一个问题,即当其权重和用户的评分数目是正相关关系时,会导致评分数目多的项目受到更大的影响,此时应该使用一个影响权重进行处理,其可以表示为如下公式:

$$f(i) = \begin{cases} 1 & (\text{user}(i) \geq \text{average} * 0.5) \\ \frac{\text{user}(i)}{0.5 * \text{average}} & (\text{user}(i) < \text{average} * 0.5) \end{cases} \quad (10)$$

在传递相似度时,两个物品 i, j 的信任关系即为:

$$T_{i \rightarrow j} = \frac{\text{SuccessSet}(i, j)}{\text{RecommendSet}(i, j)} * f(i) \quad (11)$$

1.4 相似度传递的计算

由于数据中往往存在明显的稀疏问题,将会影响到相似度的计算结果,因此需要通过对相似度的传递来解决此问题。在这个过程中要利用项目之间的信任关系进行分析,其具体的过程如下所示:首先根据信任矩阵得到信任度最高的项目集合,然后根据步骤 1 中的结果得到项目间的间接相似关系。

首先需要根据信任矩阵得到项目 i 的信任集合,并且已知 p 属于 I ,然后求解两个项目 i, j 的间接相似度,其中传递时的权值是项目 i 对项目 p 的信任关系,最后采用加权平均的方式计算间接相似 $\text{indirect}_{\text{sim}(i, j)}$,其公式如下所示:

$$\text{indirect}_{\text{sim}(i, j)} = \frac{\sum_{p \in I} T_{i \rightarrow p} * \text{sim}(p, j)}{\sum_{p \in I} T_{i \rightarrow p}} \quad (12)$$

其中直接相似度可以采用余弦相似度方法进行计算,而间接相似度的计算过程较为复杂,其需要先根据项目之间的信任关系进行建模,在此基础上传递直接相似度。整个算法可以划分为如下多个过程:

- (1) 根据用户项目评分矩阵对其相似度进行计算;
- (2) 根据用户项目评分矩阵来构建可信关系模型;
- (3) 基于上一步得到的可信关系模型来传递相似度,并得到其间接相似度;
- (4) 根据得到的直接与间接相似度进行加权计算,可以计算出项目间的相似度;
- (5) 根据相似度获取项目的最近项目邻居集合;
- (6) 可以对未评分项目进行评分。

2 隐语义模型

2.1 定义

隐语义模型^[11]是一种利用矩阵分解得到结果的算法,实质就是通过降维的方法将一些没有用的信息和噪声剔除,从而提高预测数据的准确性,也在一定程度上缓和了数据稀疏带来的不良影响。LFM 当属矩阵分解算法中最适合推荐系统的模型。

Simon Funk^[12]将高维矩阵进行分解,得到维数较低的两个矩阵的乘积,如下所示:

$$R = P^T Q \quad (13)$$

其中, P 是 m 行 k 列的矩阵, Q 则为 n 行 k 列的矩阵, m 和 n 则分别代表用户的数量与项目的数量。则用户 u 对项目 i 的评分可以表示为:

$$r_{ui} = \sum_k p_{uk} * q_{ik} \quad (14)$$

可见,隐语义模型的思想本质上是借助最小化的评价方法 RMSE^[13-15] 对矩阵 P 、 Q 学习的一个过程,其中 $p_{uk} = P(u, f)$, $q_{ik} = Q(i, f)$ 。

2.2 用户兴趣迁移模型

隐语义模型着重强调的是最小化求解原则,用户对项目的评分是在保持条件不发生变化的理想条件下的,但是事实情况是,用户的兴趣有一个变化的过程,这在传统的隐语义模型中并没有得到体现。

在进行样本训练的过程中发现,与推荐时间最接近时,用户对项目的评分最能反映出用户的喜爱程度。为了比较客观地反映这种喜好,则需要将每个时间段的评分按照时间由远到近的顺序选取一个相应的权重系数,就可以很好地反映出用户的爱好在这段时间里的一个变化过程^[16]。在面对这种问题时,需要将时间因素纳入到整个评分计算公式中,作为推荐算法的一个部分,可以更好地反映出哪些更加合适用户的个人喜好。这其中的一个典型模型就是基于遗忘的兴趣模型。目前,心理学家普遍认为,喜好和记忆遵循着相似的变化规律,即随着时间向后推移而慢慢弱化,直至被人遗忘,并且,越到最后遗忘的越慢,最终会达到一种稳定状态。世界文明的一个心理学家—艾宾浩斯,对人类的遗忘现象做过一个数学统一,得到一条遗忘特性曲线,这就是著名的艾宾浩斯遗忘曲线^[17],如图1所示。

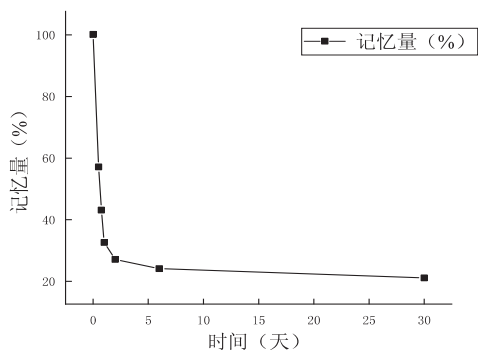


图1 艾宾浩斯遗忘曲线

接着,便逐渐开始有学者将这种规律运用到推荐算法的模型中。Koychev 等人^[18] 利用推荐系统将用户的喜好漂移导入了兴趣模型中,在他们看来,人们喜好的变化会按照艾宾浩斯曲线规律发展,因此,他们提出了一个遗忘规律性的兴趣变化模型。这种模型在记忆曲线的基础上对不同项的评分分配一个比重系数,并且设置一个阈值,如果权重的取值低于这个阈值就

忽略不计。Maloof^[19] 在此基础上又提出了一种基于遗忘窗口的兴趣变化模型,在这种模型中,用户的评分会随着权重的推移而慢慢减少。

但是在实际的推算中发现一个问题,那就是直接将记忆曲线导入推荐系统中会导致与推荐时间接近的兴趣值被过分放大,反之,较远的兴趣值会被基本忽略。对于上述问题,需要改进一下函数,重新建立一个模型,如下表示:

$$f_{u,i} = \alpha + (1 - \alpha) * e^{-(t_{now} - t_{u,i})} \quad (15)$$

其中, α 是一个 0-1 之间的变量,表示兴趣变化的影响范围, t_{now} 是现在的时间到初始时间之间的差值, $t_{u,i}$ 是用户 u 进行项目评分时的时间和初始时间的差值,初始时间的选取一般选择某个时间点作为参考点。

通过对该函数的定义进行简单分析可知,用户对项目的这种爱好程度会推着时间的推移而呈现出一个衰减的状态。在推荐时间的节点处,函数取得最大值,当时间趋向无穷远处时,函数取得最小值,无限接近 α 。关于 α 的取值,1 代表的是它将用户的评分同等看待,这和传统算法是完全一致的,若 α 的取值是 0,则意味着用户喜好的变化遵循艾宾浩斯曲线的规律特性。可见,通过对 α 的值进行调节,可以找到一个合适的值,使得用户的兴趣喜好可以很好地与评分匹配,更能满足用户的客观需求。

2.3 引入时间参数的隐语义模型

把含有用户兴趣变化的函数引入到改进的模型中,改进之后的 RMSE 可以更好地对样本集进行训练,以此来求出最佳的 P 、 Q 。所以,改进后模型的损失函数重新进行如下定义:

$$\begin{aligned} \text{CostFunction} = & \sum_{\substack{u \in U \\ i \in I}} f(t_{u,i}) * (r_{ui} - \hat{r}_{ui})^2 = \\ & \sum_{\substack{u \in U \\ i \in I}} (\alpha + (1 - \alpha) * e^{-(t_{now} - t_{u,i})}) * (r_{ui} - \sum_{k=1}^k p_{uk} * q_{ik})^2 \end{aligned} \quad (16)$$

为了将该损失函数最小化,需要通过梯度下降方法^[20] 将其做最优化处理。函数在迭代的过程中,经历一段时间的运算,便会到达一种稳定的状态,为此,可以设置一个合适的阈值,当迭代计算到这一步时,就可以停止继续迭代。

文中算法先设定一个数值 K 来判断该函数是否需要继续运算,通过梯度下降的方法进行求解的过程如下所示:

(1) 设定两个参数变量 α 和 K ,前者表示迭代的步长,而后者表示需要进行迭代的次数。需要指出的是,随着迭代过程的推进,步长会慢慢的缩减。

(2) 计算当前节点的梯度值 $\nabla \text{CostFunction}(p_{uk},$

q_{ik}), 需要对 p_{uk}, q_{ik} 分别进行求偏导, 结果如下:

$$\begin{aligned}\frac{\partial C}{\partial p_{uk}} &= -2q_{ik} * e_{ui} \\ \frac{\partial C}{\partial q_{ik}} &= -2p_{uk} * e_{ui}\end{aligned}\quad (17)$$

其中, e_{ui} 代表的是 $(r_{ui} - \hat{r}_{ui})$ 。

使用下面的公式进行迭代计算, 刷新 p_{uk} 和 q_{ik} 的数值。

$$\begin{aligned}p_{uk} &= p_{uk} + \alpha * q_{ik} * e_{ui} \\ q_{ik} &= q_{ik} + \alpha * p_{uk} * e_{ui}\end{aligned}\quad (18)$$

(3) 记录迭代计算的次数与 K 值进行比较, 当迭代次数超过 K 时, 停止迭代计算, 反之再进入第二步。

一般情况下, 函数 CostFunction 在迭代过程中一定会找到一个极值, 如果不出意外的话, 这一点就是要求的最小值。

3 混合推荐算法

尝试着将以上两种算法整合到一起, 既可以解决降维处理时的数据丢失问题, 也可以在一定程度上缓和稀疏问题。时间函数的引入可以很好地反映出用户的喜好兴趣的变化, 更有助于提高系统的推荐性。在融合时, 采用将两种算法隔离开来同时运行, 得到的运算结果再进行加权求和的方式。

具体的计算公式如下:

$$P_{ui} = \sum_k p_{uk} * q_{ik} + \frac{\sum_{j \in I} \text{sim}(i, j) * r_{uj}}{\sum_{j \in I} \text{sim}(i, j)} \quad (19)$$

传统意义上的协同算法, 计算相似度时, 主要借助的是相关系数法。文中拟采用线性回归法进行求解, 为此, 需要改写上面的计算公式:

$$P_{ui} = \sum_k p_{uk} * q_{ik} + \sum_{j \in I} w_{ij} * r_{uj} \quad (20)$$

其中, $w_{ij} = \frac{\text{sim}(i, j)}{\sum_{j \in I} \text{sim}(i, j)}$ 。借助这个公式, 可以很

方便地进行参数的计算求解, 至于计算的过程, 则采取的是上一章中的时间函数权重法, 将 RMSE 进行最小化处理。

4 实验结果与分析

4.1 实验数据集

本节将通过实验的方式对算法的具体效果进行验证。在进行测试时首先需要获取实验数据集, 这里采用的是 MovieLens^[21]。MovieLens 数据集含有的用户数目与电影数目分别是 968 和 1 762, 用户评分范围是在 1~5 之间, 各个用户已经评分的电影数目都高于 1/5; 为了有效地对算法的效果进行验证, 需要将所有数据划分为训练集与测试集, 其比例分别是 70% 与

30%。将邻居集合数目设置为多个不同的值。

RMSE^[22-23] 在推荐系统的评价体系中占据着非常重要的位置, 是衡量系统性能的一个不可或缺的指标。一般的推荐系统会在用户登录系统的那一天就开始推荐, 但是因为用户兴趣的变化不断, 理论上, 和推荐当天距离最短的兴趣就越有可能被推荐为当天的兴趣, 在 RMSE 中将这此兴趣点所对应的时间的比例增加, 对模型中的 RMSE 做了如下修改:

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i \in T} f_{u,i} * (r_{ui} - \hat{r}_{ui})^2}{|T|}} \quad (21)$$

其中, r_{ui} 表示用户 u 对项目 i 的评分, \hat{r}_{ui} 表示用户 u 对项目 i 的预测评分, $|T|$ 表示评分正确的总数, $f_{u,i}$ 表示时间函数。

4.2 实验结果

(1) 参数 α 值对 RMSE 的影响。

参数 α 的作用就是控制时间造成的用户兴趣的变化, 同时会以最小权重的方式设置下限值。事实上, 用户喜好的变化是有一定的规律可循的, 因此, 需要设置不同的时间权重进行数次实验。将邻居集合数目设置为 100, 迭代次数为 200, 学习速率为 0.05, 实验得到的 RMSE 数值如图 2 所示。

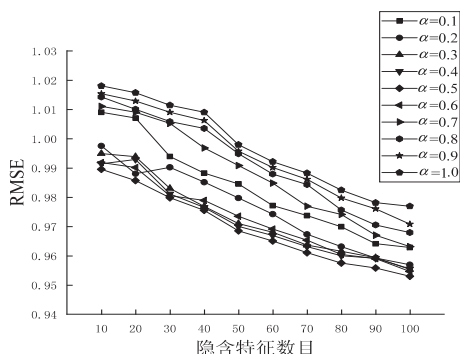


图 2 不同参数 α 下的 RMSE

从图 2 可以看出, 在 α 值变化的过程中, RMSE 的数值也会相应地发生变化。当 α 的取值是 0.5 时, 得到的结果是最好的。

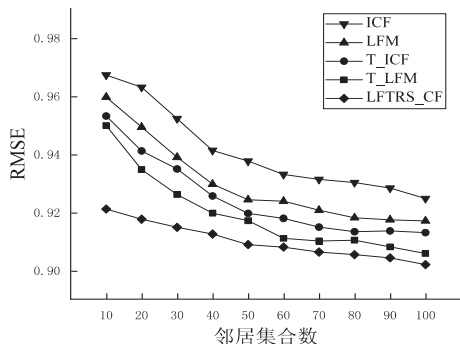


图 3 不同邻居个数下各融合策略下的 RMSE 比较

(2) 多种算法对 MovieLens 数据集的测试。

采用多种算法对数据集进行了测试, 设置不同的

邻居集合,得到了不同邻居集合下的 RMSE,如图 3 所示。

通过图 3 能够明显看到,提出的 LFTRS_CF 算法的 RMSE 值最小,说明其在准确性方面能够达到较好的效果,优于其他的四种算法。

5 结束语

文中详细研究了推荐算法的优化策略,提出了两种算法,基于项目相似度的协同过滤算法和基于用户兴趣迁移的隐语义模型算法,并在此基础上对其进行线性融合,使得其中存在的数据稀疏性问题和用户兴趣迁移问题得到了较好的解决。提出的混合推荐算法既可以对丢失的信息进行补充,较好地适应用户兴趣的变化,同时大大弱化了数据的稀疏导致的一系列负面影响。

参考文献:

- [1] 项亮. 推荐系统实践[M]. 北京:人民邮电出版社,2012.
- [2] SHI Y, LARSON M, HANJALIC A. Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges[J]. ACM Computing Surveys, 2014, 47(1): 1-45.
- [3] WANG J, DE VRIES A P, REINDERS M J T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion[C]//Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. Seattle, Washington, USA: ACM, 2006.
- [4] LINDEN G, SMITH B, YORK J. Amazon. com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [5] 王国霞, 刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用, 2012, 48(7): 66-76.
- [6] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [7] MANOGARAN G, VARATHARAJAN R, PRIYAN M K. Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system[J]. Multimedia Tools and Applications, 2018, 77(4): 4379-4399.
- [8] WANG Z, LIAO J, CAO Q, et al. Friendbook: a semantic-based friend recommendation system for social networks[J]. IEEE Transactions on Mobile Computing, 2015, 14(3): 538-551.
- [9] WANG W, JING Y, LIANG H. An improved collaborative filtering based on item similarity modified and common ratings[C]//2012 international conference on cyberworlds. Darmstadt: IEEE, 2012: 231-235.
- [10] XIANG L, YANG Q. Time-dependent models in collaborative filtering based recommender system[C]//2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology. Milan: IEEE, 2009: 450-457.
- [11] 肖文强, 姚世军, 吴善明. 一种改进的 top-N 协同过滤推荐算法[J]. 计算机应用研究, 2018, 35(1): 105-108.
- [12] WANG Y, SUN H, ZHANG R. AdaMF: adaptive boosting matrix factorization for recommender system[C]//International conference on web-age information management. [s. l.]: [s. n.], 2014: 43-54.
- [13] 于金明, 孟军, 吴秋峰. 基于改进相似性度量的项目协同过滤推荐算法[J]. 计算机应用, 2017, 37(5): 1387-1391.
- [14] ZHOU X, SHU W, LIN F, et al. Confidence-weighted bias model for online collaborative filtering[J]. Applied Soft Computing, 2017, 70: 1042-1053.
- [15] YANG C, YU X, LIU Y, et al. Collaborative filtering with weighted opinion aspects[J]. Neurocomputing, 2016, 210: 185-196.
- [16] 陈天昊, 帅建梅, 朱明. 一种基于协作过滤的电影推荐方法[J]. 计算机工程, 2014, 40(1): 55-58.
- [17] 孙海峰, 甘明鑫, 刘鑫, 等. 国外电影推荐系统网站研究与评述[J]. 计算机应用, 2013, 33(s2): 119-124.
- [18] ZHANG C, ZHAO X, WANG K, et al. Content+attributes: a latent factor model for recommending scientific papers in heterogeneous academic networks[C]//European conference on information retrieval. Amsterdam: Springer, 2014: 39-50.
- [19] QIU L, GAO S, CHENG W, et al. Aspect-based latent factor model by integrating ratings and reviews for recommender system[J]. Knowledge-Based Systems, 2016, 110: 233-243.
- [20] ZENG X, DING N, QUAN Z. Latent factor model with heterogeneous similarity regularization for predicting gene-disease associations[C]//2016 IEEE international conference on bioinformatics and biomedicine. Shenzhen: IEEE, 2016: 15-18.
- [21] CHENG Z, DING Y, ZHU L, et al. Aspect-aware latent factor model: rating prediction with ratings and reviews[C]//International world wide web conference committee. Lyon: ACM, 2018: 23-27.
- [22] LUO X, ZHOU M C, LI S, et al. A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method[J]. IEEE Transactions on Neural Networks & Learning Systems, 2016, 27(3): 579-592.
- [23] CHENG J, YUAN T, WANG J, et al. Group latent factor model for recommendation with multiple user behaviors[C]//Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval. Gold Coast: ACM, 2014: 995-998.