用户间多相似度协同过滤推荐算法

范 波 程久军
(同济大学计算机科学与工程系 上海 201804)
(同济大学嵌入式系统与服务计算教育部重点实验室 上海 201804)

摘 要 传统的User-based协同过滤推荐算法仅采用了单一的评分相似度来度量用户之间对任何项目喜好的相似程度。然而根据日常经验,人们对不同类型事物的喜好程度往往是不同的,单一的评分相似度显然无法准确描述这种不同。针对上述问题,提出了一种基于用户间多相似度的协同过滤推荐算法,即基于用户间对不同项目类型的多个评分相似度来计算用户对未评分项目的预测评分。实验结果表明,该算法可以有效地提高预测评分的准确性及推荐质量。

关键词 多相似度,协同过滤推荐算法,User-based,MAE
中图法分类号 TP391 **文献标识码** A

Collaborative Filtering Recommendation Algorithm Based on User's Multi-similarity

FAN Bo CHENG Jiu-jun
(Department of Computer Science & Engineering, Tongji University, Shanghai 201804, China)
(Key Laboratory of Embedded System and Service Computing of Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract Conventional user-based collaborative filtering algorithm measures the similarity of two user's favor of any types of items through the single rating similarity. However, daily experience tells us that people usually have different degree on their favor of different types of objects, and obviously the single rating similarity cannot accurately describe this difference. Aiming at this problem, we deeply analyzed the characteristic of user-based collaborative filtering recommendation algorithm, and proposed a collaborative filtering recommendation algorithm based on user's multi-similarity, which describes different similarity of two user's favor of different types of items through the computation of their multiple independent rating similarity for different types of items. The experimental results show that the proposed algorithm, which computes predicted ratings of unrated items on the basis of user's multi-similarity, can effectively improve the accuracy of predicted ratings and enhance the quality of recommendation.

Keywords Multi-similarity, Collaborative filtering recommendation algorithm, User-based, MAE

1 引言

互联网信息技术的飞速发展,极大地改变了人们获取信息的方式。然而互联网信息总量的日益激增,也使得信息过载问题日益严重。作为一种能有效解决互联网信息过载问题的技术手段,推荐系统正越来越受到业界的重视。推荐系统通过学习用户的行为,了解和掌握用户的偏好,从而可以更有针对性地向用户推荐他们可能感兴趣的内容。目前互联网的几大支柱产业,包括电子商务和社交网络,均不同程度地使用了推荐系统技术。

在已有的推荐算法中,协同过滤推荐算法[1]是比较成熟,也是在实际应用中使用非常广泛的一类推荐算法。目前的大部分协同过滤推荐算法,主要是通过计算某一用户对未评分项目的预测评分并以此作为主要依据来向该用户进行推荐。其中,传统的User-based协同过滤推荐算法[2]将预测评分的计算建立在与当前用户相似的其他用户的真实评分记录的基础上。何谓"相似"的用户,文献[3]认为是指"兴趣和口味相似"的用户。因此,相似的用户在给项目评分时往往会给出相近的评分结果,而这也是利用协同过滤的思想进行评分预测的主要依据。

为了进一步提高User-based协同过滤推荐算法的推荐质量,针对该算法存在的一些缺陷,比如新用户问题、新项目问题[4],以及数据的稀疏问题[4]和可扩展性[5]等问题,许多学者提出了一些新的方法来对该算法进行改进。比如,文献[4]提出,通过融合协同过滤推荐算法和基于内容的推荐算法[3,4],可以有效缓解新用户问题和新项目问题带来的影响;文献[6]提出通过缺省投票(Default Voting)、倒排用户频率(Inverse User Frequency)等方法缓解数据稀疏问题;文献[5]基于项目之间的相似性相对稳定的观察,提出了Item-based协同过滤推荐算法,从而提高了系统的可扩展性;文献[7-9]

到稿日期:2011-04-21 返修日期:2011-07-18 本文受国家自然科学基金项目(90818023),上海市项目(08GG08,09JC1414200),教育部新教师基金(20090072120048)资助。

范 波(1987—),男,硕士生,主要研究方向为推荐系统,E-mail:fanb093@163.com;程久军 男,副教授,主要研究方向为移动计算、物联网等。

在文献[5]的基础上，进一步融合了 User-based 和 Item-based 协同过滤推荐算法，有效地解决了数据稀疏的问题，并提高了算法的推荐质量。

然而，User-based 协同过滤推荐算法本身，以及目前已知的基于该算法的改进，都仅仅采用了单一的相似度来描述用户之间对于任何项目喜好的相似程度，而没有考虑到项目所属类型的不同对相似度产生的影响。然而日常经验告诉我们，人们对于不同类型事物的喜好程度往往是不同的。也就是说，同时对某类事物感兴趣的两个人，对于另一类事物的喜好程度有可能会完全不同。表现在对项目的评分上就是这两人对于某一类型的项目的评分比较相近，但是对于另一类项目的评分则有可能大相径庭。因此，从这个角度来看，传统的 User-based 协同过滤推荐算法仅依靠单一的相似度来度量两个用户对于所有类型项目喜好的相似程度并不是很合理。

针对上述问题，本文提出了一种基于用户间多相似度的协同过滤推荐算法，即通过计算用户间对不同项目类型的多个独立的评分相似度，来分别描述用户间对不同类型项目喜好的相似程度，从而更加准确地刻画用户之间对不同项目类型的真实偏好，并在此基础上有效地提高预测评分的准确性及推荐质量。

2 传统的 User-based 协同过滤推荐算法

传统的 User-based 协同过滤推荐算法可以分成 3 个阶段：(1) 根据用户的历史评分记录计算用户之间的评分相似度；(2) 从与当前用户相似度最高的用户中选取若干个作为最近邻，根据这些最近邻对于某一项目的实际评分来预测当前用户对该项目的评分；(3) 选取预测评分最高的若干个项目作为推荐结果提供给当前用户。而目前对 User-based 协同过滤推荐算法的研究和改进，主要集中在前两个阶段，即用户相似度以及预测评分的计算。

2.1 用户相似度

为了计算用户对未评分项目的预测评分，首先需要根据用户的历史评分记录计算用户之间的评分相似度。相似度体现了用户之间对于各项目喜好的相似程度。相似度通常有两种计算方法[5,6]：相关相似性和余弦相似性。

(1) 相关(Correlation)相似性：在相关相似性的计算中，根据两个用户共同评分过的所有项目的评分来计算 Pearson 相关系数[10]，并以此作为两个用户之间的相似度。假设两个用户 x、y 共同评分过的项目集合为 $I_{x,y}$；$r_{x,i}$ 和 $r_{y,i}$ 分别表示用户 x 和 y 对项目 i 的评分；$\overline{r_x}$ 和 $\overline{r_y}$ 分别表示 x 和 y 对所有项目评分的平均分。则用户 x、y 之间的相关相似性为

$$\mathrm{Sim}(x,y)=\frac{\sum_{i\in I_{x,y}}(r_{x,i}-\overline{r_x})(r_{y,i}-\overline{r_y})}{\sqrt{\sum_{i\in I_{x,y}}(r_{x,i}-\overline{r_x})^2}\sqrt{\sum_{i\in I_{x,y}}(r_{y,i}-\overline{r_y})^2}} \tag{1}$$

(2) 余弦(Cosine)相似性：余弦相似性也叫向量相似性[6]。每个用户的所有评分记录被看成 n 维项目空间中的一个向量(如果用户没有对某个项目打分，则该用户对该项目的评分被设为零)，两个用户之间的相似度被看成这两个用户的评分向量之间夹角的余弦。假设两个用户 x、y 的评分向量分别为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$，集合 I_x、I_y 分别表示 x 和 y 评分过的项目集合，则他们之间的余弦相似性为

$$\mathrm{Sim}(x,y)=\cos(\boldsymbol{x},\boldsymbol{y})=\frac{\sum_{i\in I_{x,y}}r_{x,i}r_{y,i}}{\sqrt{\sum_{i\in I_x}r_{x,i}^2}\sqrt{\sum_{i\in I_y}r_{y,i}^2}} \tag{2}$$

2.2 预测评分

预测评分体现了当前用户对于该项目可能的喜好程度，它的计算建立在与当前用户相似的其他用户对于该项目的真实评分之上。预测评分是大多数推荐系统产生推荐结果的主要依据。

在计算预测评分时，经常采用 K 近邻[11]方法，也就是选择与当前用户相似度最高的 K 个用户作为当前用户的邻居节点来进行计算。设集合 N_u 表示用户 u 的最近邻集合，为了计算用户 u 对项目 i 的预测评分，经常采用如下的计算公式。

$$R(u,i)=\overline{r_u}+\frac{\sum_{u'\in N_u}\mathrm{Sim}(u,u')(r_{u',i}-\overline{r_{u'}})}{\sum_{u'\in N_u}\mathrm{Sim}(u,u')} \tag{3}$$

3 基于用户间多相似度的协同过滤推荐算法

传统的 User-based 协同过滤推荐算法仅依靠单一的评分相似度来计算用户对于未评分项目的预测评分，这样做并不合理。从本文第 2 节可以看到，这个单一的评分相似度是基于两个用户所有的历史评分记录计算得出的，而没有考虑到项目所属的类型，因此这个相似度仅仅反映了这两个用户之间所有历史评分记录的相似程度，而无法反映出他们对于某一类型项目的真实喜好的相似程度，因此在一定程度上降低了算法的推荐质量。

在现实生活中，人们对于不同类型事物的喜好程度往往是不同的。正是基于这种观察，本文提出了基于用户间多相似度的协同过滤推荐算法。用户间的多相似度，就是指用户之间关于不同项目类型的多个独立的评分相似度。用户之间关于某一项目类型的相似度是基于这两个用户对于属于该类型的项目的评分记录计算得到的，因此这个相似度可以更加准确地反映出这两个用户对于该类型项目的喜好的相似程度。这样通过分别计算用户之间对于不同项目类型的各自独立的评分相似度，便可利用多个相似度更加准确地刻画出用户之间对于各种类型项目的兴趣和口味的异同，并在此基础上得到更加准确的预测评分。

这里，设集合 $T=\{t_1,t_2,\cdots,t_k\}$ 表示所有类型的集合，每个项目可以属于集合 T 中的一种或多种类型。为了描述用户对于这 k 种类型的项目不同的喜好程度，需要计算用户间基于所有 k 种项目类型的相似度。对于任意两个用户 x 和 y，他们关于集合 T 中各个类型的 k 个相似度表示为 $\mathrm{Sim}(x,y,t_1)$，$\mathrm{Sim}(x,y,t_2)$，$\cdots$，$\mathrm{Sim}(x,y,t_k)$。

3.1 用户间的多相似度计算

对于用户间多相似度的计算，也可以采用相关相似性或余弦相似性进行计算，区别在于在计算用户间基于某种项目类型的相似度时，只用到用户对于属于该类型的项目的评分记录。

设用户 x 和 y 共同评分过的属于类型 t_m ($t_m\in T$，$1\leqslant m\leqslant k$) 的项目的集合为 $I_{x,y}^{t_m}$，$\overline{r_x^{t_m}}$，$\overline{r_y^{t_m}}$ 分别表示 x 和 y 对于属于类型 t_m 的项目的平均评分，根据式(1)，用户 x 和 y 关于类型 t_m 的相关相似度为

$$\mathrm{Sim}(x,y,t_m)=\frac{\sum_{i\in I_{x,y}^{t_m}}(r_{x,i}-\overline{r_x^{t_m}})(r_{y,i}-\overline{r_y^{t_m}})}{\sqrt{\sum_{i\in I_{x,y}^{t_m}}(r_{x,i}-\overline{r_x^{t_m}})^2}\sqrt{\sum_{i\in I_{x,y}^{t_m}}(r_{y,i}-\overline{r_y^{t_m}})^2}} \tag{4}$$

设集合 $I_x^{t_m}$ 和 $I_y^{t_m}$ 分别表示用户 x 和 y 评分过的属于类型 t_m 的项目的集合，他们共同评分过的属于类型 t_m 的项目的集合为 $I_{x,y}^{t_m}$，根据式(2)，用户 x 和 y 关于类型 t_m 的余弦相似度为

$$\mathrm{Sim}(x,y,t_m)=\frac{\sum_{i\in I_{x,y}^{t_m}} r_{x,i}r_{y,i}}{\sqrt{\sum_{i\in I_x^{t_m}} r_{x,i}^2}\sqrt{\sum_{i\in I_y^{t_m}} r_{y,i}^2}} \tag{5}$$

可以看到，式(4)和式(5)形式上分别与式(1)和式(2)类似，但是由于只用到了用户对于属于该类型的项目的评分记录，因此计算得到的是两个用户关于该类型的评分相似度。通过采用用户对于不同类型项目的评分记录分别进行计算，就可以得到用户间关于不同项目类型的各自独立的评分相似度。

3.2 基于用户间多相似度的预测评分计算

在计算基于用户间多相似度的预测评分时，同样采用 K 近邻方法。与第 2 节介绍的传统的 User-based 协同过滤推荐算法不同，由于考虑到了用户间对于不同类型的项目有不同的相似度，而通常一个项目可以属于多个类型，因此，在计算当前用户对于某一项目的预测评分时，首先需要从其他用户中分别选取 K 个与当前用户关于该项目所属的各个类型的相似度最高的用户作为最近邻，来分别计算该用户基于该项目所属的各个类型的多个预测评分，最后取它们的加权平均作为最后得到的预测评分。

这里，我们选取用户 u 对于项目 i 的预测评分的计算过程。设 i 所属类型的集合为 $T_i=\{t_{i1},t_{i2},\cdots,t_{is}\}$，则 $T_i\subseteq T$，其中 T 是所有类型的集合。对于类型 $t_{ij}\in T_i$(其中 $1\leqslant j\leqslant s$)，设用户关于类型 t_{ij} 的 K 最近邻用户集合为 $N_u^{t_{ij}}$，根据式(3)，由 $N_u^{t_{ij}}$ 计算得到的用户 u 对项目 i 的预测评分为

$$R(u,i,t_{ij})=\overline{r_u^{t_{ij}}}+\frac{\sum_{u'\in N_u^{t_{ij}}}\mathrm{Sim}(u,u',t_{ij})(r_{u',i}-\overline{r_{u'}^{t_{ij}}})}{\sum_{u'\in N_u^{t_{ij}}}\mathrm{Sim}(u,u',t_{ij})} \tag{6}$$

以此类推，分别得到 s 个预测评分 $R(u,i,t_{i1})$，$R(u,i,t_{i2})$，$\cdots$，$R(u,i,t_{is})$，最后取它们的加权平均值作为最后的预测评分

$$R(u,i)=\sum_{j=1}^{s}\lambda_j R(u,i,t_{ij}) \tag{7}$$

式中，$\lambda_j=|I_u^{t_{ij}}|/|I_u^{T_i}|$，$|I_u^{t_{ij}}|$ 表示用户 u 评分过的属于类型 t_{ij} 的项目的数量，$|I_u^{T_i}|$ 表示用户 u 评分过的属于类型集 T_i 中的任意一种类型的项目的数量。很显然，有

$$\sum_{j=1}^{s}\lambda_j=1 \tag{8}$$

权重 λ_j 反映了用户对于类型 t_{ij} 在项目 i 所属类型的集合 T_i 中的关注程度，λ_j 值越高，则用户对于属于类型 t_{ij} 的项目的评分次数相对于用户对属于集合 T_i 中的其他类型的项目的评分次数的比重就越高，因此根据类型 t_{ij} 得出的预测评分 $R(u,i,t_{ij})$ 更有可能反映出用户对于项目 i 真实的喜好程度。通过抬高 $R(u,i,t_{ij})$ 的权重，可以得到更加准确的预测评分。

3.3 算法过程及时间复杂度分析

如前所述，对于 User-based 协同过滤推荐算法的研究主要集中在用户相似度计算和预测评分计算两个阶段。因此，下面就本文提出的基于用户间多相似度的协同过滤推荐算法，分别给出这两个阶段的算法过程描述以及时间复杂度的分析。

算法 1 用户间的多相似度计算

输入：用户集合 U、项目集合 I、项目类型集合 T、评分矩阵 $R_{U\times I}$

输出：用户间的多相似度矩阵 $\mathrm{Sim}_{U\times U\times T}$

Step 1 设 $i=1$；

Step 2 若 $i==|U|$，结束，否则跳转至 Step 3；

Step 3 设 $j=i+1$；

Step 4 若 $j>|U|$，$++i$，跳转至 Step2，否则跳转至 Step 5；

Step 5 对于 $\forall t\in T$，分别执行以下两个步骤；

Step 5.1 从评分矩阵 R 中分别选择用户 u_i 和 u_j 对属于类型 t 的项目的评分记录；

Step 5.2 根据式(4)或式(5)计算用户 u_i 和 u_j 关于类型 t 的相似度 $\mathrm{Sim}(u_i,u_j,t)$；

Step 6 $++j$，跳转至 Step 4。

可以看出，在计算用户间的多相似度时，影响时间复杂度的主要是用户的数量、项目的数量以及项目的类型数量。设用户数量为 $|U|$，项目数量为 $|I|$，项目类型数量为 $|T|$。由式(4)和式(5)可知，计算某两个用户之间关于某个项目类型的相似度的时间复杂度为 $O(|I|)$，因此两个用户之间关于所有 $|T|$ 种项目类型的时间复杂度为 $|T|\times O(|I|)=O(|T|\cdot|I|)$。最后，由于需要对任意两个用户关于 $|T|$ 种项目类型的相似度都进行计算，因此总的时间复杂度为 $\frac{|U|\times(|U|-1)}{2}\times O(|T|\cdot|I|)=O(|U|^2\cdot|T|\cdot|I|)$。

算法 2 预测评分计算

输入：目标用户 u、目标项目 i、选取的近邻数量 K、评分矩阵 $R_{U\times I}$、用户间的多相似度矩阵 $\mathrm{Sim}_{U\times U\times T}$；

输出：用户 u 对项目 i 的预测评分 $R(u,i)$；

Step 1 对于 $\forall t\in T_i$(集合 T_i 表示项目 i 所属的类型集合)，分别执行以下两个步骤；

Step 1.1 从矩阵 $\mathrm{Sim}_{U\times U\times T}$ 中选择 K 个与用户 u 关于类型 t 的相似度最高的用户组成最近邻集合 N_u^t；

Step 1.2 根据式(6)计算用户 u 对项目 i 关于类型 t 的预测评分 $R(u,i,t)$；

Step 2 根据式(7)计算用户 u 对项目 i 最后的预测评分 $R(u,i)$。

在计算用户对于某一项目的预测评分时，影响时间复杂度的主要是与当前用户相似度最高的近邻用户数量以及项目类型的数量。设每次计算预测评分时选取的近邻用户的数量为 K，项目类型的数量为 $|T|$，由式(6)和式(7)可知，计算用户对于某一项目的预测评分的复杂度为 $|T|\times O(K)=O(K\cdot|T|)$。

4 实验结果及分析

为了检验基于用户间多相似度的协同过滤推荐算法与传统的 User-based 协同过滤推荐算法之间推荐质量的差别，本文采用明尼苏达大学的 GroupLens 小组提供并维护的 MovieLens[12] 数据集进行实验。针对相关相似性和余弦相似性两种不同的用户相似性度量方法，分别进行了两组实验，每组实验分别从数据集中选择 100、500 和 1000 个用户的评分记录作为测试数据集，并进一步在这个测试数据集中随机抽取 80% 的评分记录作为训练集，另外 20% 作为测试集。实验过程中，我们采用平均绝对偏差 MAE[5] (Mean Absolute Error) 作为评价推荐算法的标准，近邻数量从 5 取到 30，间隔为 5。实验结果分别如图 1 和图 2 所示。

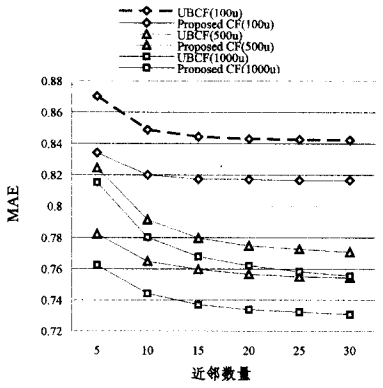

图 1　基于相关相似性的实验结果

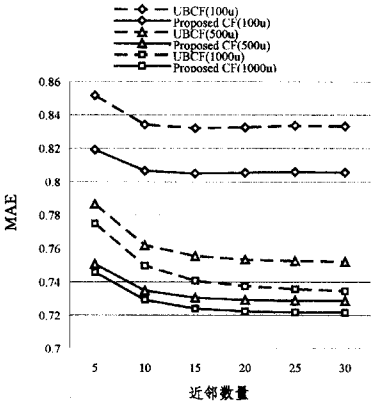

图 2　基于余弦相似性的实验结果

从图 1 可以看出，当采用 100 个用户的数据集进行实验时，在采用相关相似性计算用户间相似度的情况下，随着近邻数量的增加，无论是 User-based 协同过滤推荐算法，还是基于用户间多相似度的协同过滤推荐算法，它们的 MAE 值都呈现出下降趋势，但是无论选取的近邻数量是多少，在近邻数量相同的前提下，基于用户间多相似度的协同过滤推荐算法均具有较小的 MAE 值。同样，在 500 个用户以及 1000 个用户的数据集上进行的实验也获得了类似的结果。这说明在基于相关相似性计算用户间相似度的情况下，相比传统的 User-based 协同过滤推荐算法，采用基于用户间多相似度的协同过滤推荐算法可以得到更好的推荐效果。

同时，通过纵向比较 100 个用户、500 个用户和 1000 个用户的实验结果可以看出，两个算法在近邻数量相同的情况下，MAE 值都随着数据集中用户数量的增加而下降，这说明用户数量的增加可以有效提高推荐质量。

从图 2 可以看出，在采用余弦相似性计算用户间相似度的情况下，也可以得出和图 1 类似的结论。这说明无论采用相关相似性还是余弦相似性度量方法，在选取的近邻数量相同的情况下，基于用户间多相似度的协同过滤推荐算法的推荐效果均优于传统的 User-based 协同过滤推荐算法。

此外，从图 1 和图 2 可以看出，无论是 User-based 协同过滤推荐算法，还是基于用户间多相似度的协同过滤推荐算法，它们的 MAE 值曲线的下降速度都随着近邻数量的增加而逐渐减小。由此我们可以推断出，过多的近邻数量并不会明显地提高推荐算法的推荐质量，而近邻数量的增加势必会造成运算量的增加，因此应当做到在保证推荐质量的前提下，尽量选择较少的近邻来进行计算。

结束语　协同过滤推荐算法是推荐系统中使用得非常广泛，也是比较成熟的一类推荐算法。相对于其他类型的推荐算法，协同过滤推荐算法有其特有的优点。然而，传统的 User-based 协同过滤推荐算法只使用了单一的用户间相似度来计算用户对于某个项目的预测评分，而没有考虑到项目类型的不同对相似度产生的影响。通常来讲，用户对于不同类型项目的喜好程度是不同的，单一的相似度很难准确地描述这种不同，这将在一定程度上造成推荐质量的下降。

针对传统的 User-based 协同过滤推荐算法中的这种不足，本文提出了基于用户间多相似度的协同过滤推荐算法，即通过计算用户间对于不同项目类型的不同评分相似度，得出用户对于属于不同类型的项目的更加准确的预测评分。MovieLens 数据集上的实验，进一步表明基于用户间多相似度的协同过滤推荐算法相比传统的 User-based 协同过滤推荐算法，可以获得更好的推荐效果。

参 考 文 献

[1] Herlocker J L, Konstan J A, Borchers A, et al. An Algorithmic Framework for Performing Collaborative Filtering[C]// SIGIR 99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999: 230-237

[2] Resnick P, Iacovou N, Suchak M, et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews[C]// Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. 1994: 175-186

[3] 许海玲，吴潇，李晓东，等. 互联网推荐系统比较研究[J]. 软件学报，2009，20(2)：350-362

[4] Adomavicius G, Tuzhilin A. Towards the Next Generation of Recommender Systems: a Survey of the State-of-the-art and Possible Extensions[J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17(6): 734-749

[5] Sarwar B, Karypis G, Konstan J, et al. Item-Based Collaborative Filtering Recommendation Algorithms[C]// Proceedings of the 10th International World Wide Web Conference. New York, 2001: 285-295

[6] Breese J, Hecherman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 98). 1998: 43-52

[7] 邓爱林，朱扬勇，施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报，2003，14(9)：1621-1628

[8] 黄创光，印鉴，汪静，等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报，2010，33(8)：1369-1377

[9] Wang J, Vries A, Reinders M. Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion [C]// SIGIR 06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2006: 501-508

[10] Shardanand U, Maes P. Social Information Filtering: Algorithms for Automating 'Word of Mouth'[C]// Proceeding of the Conference on Human Factors in Computing Systems. 1995: 210-217

[11] 罗辛，欧阳元新，熊璋，等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报，2010，33(8)：1437-1445

[12] Miller B N, Albert I, Lam S K, et al. MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System [C]// IUI 03: Proceedings of the 8th International Conference on Intelligent User Interfaces. New York, 2003: 263-266