稀疏数据集协同过滤算法的进一步研究

罗 琦 缪昕杰 魏 倩
(南京信息工程大学信息与控制学院 南京 210044)

摘 要 协同过滤算法是电子商务和信息系统中非常重要的一门技术。其中用户相似度度量方法的科学性至关重要。为了获得更好的精度,采用用户间共同评分数目来动态调节原相似度,以更准确地反映用户间相似度的真实性。在此基础上,根据社会网络中FTL模型(follow the leader)的思想,对新用户或找不到最近邻的用户采用基于专家信任度的预测算法代替传统相似度来预测用户的评分,弥补了传统算法的不足。实验表明,算法提高了预测评分的准确性和推荐质量,并缓解了新用户的冷启动问题。

关键词 协同过滤,推荐算法,相似度,冷启动
中图法分类号 TP391 **文献标识码** A

Further Research on Collaborative Filtering Algorithm for Sparse Data

LUO Qi MIAO Xin-jie WEI Qian
(College of Information and Control, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract In electronic commerce and information system, collaborative filtering is a very important technique. User similarity measure of the scientific method is crucial. In order to obtain better accuracy, numbers of common Ratings between users were used to dynamically adjust the original similarity to more accuratelly reflect the authenticity of the similarity between users. On this basis, according to the social network FTL model (follow the leader) thoughts, for new users or users who cannot find the nearest neighbor, prediction algorithm based on expert trust degree was used instead of similarity to predict the user's score, making up the deficiency of the traditional algorithm. Experiments show that the algorithm can improve the prediction score, the accuracy and the quality of recommendation, and alleviate the cold-start problem for new users.

Keywords Collaborative filtering, Recommendation algorithm, Similarity, Cold start

1 引言

随着信息技术的快速发展,人们周围的数据量正在以指数级的速度增长,使得信息过载的问题日益明显。如何让人们能在最短的时间内找到他们最需要的东西成为了当今非常重要的一个研究热点。个性化推荐技术应运而生,它能够快速地发现人们的兴趣与需求。协同过滤算法[1]是个性化推荐中应用得最为广泛的一门技术。目前大部分协同过滤算法都是在用户间相似度计算的基础上,找到与之兴趣相近的用户,再利用此类用户的评分来预测目标用户未评分的项目,最后产生推荐。

然而随着数据量的不断增加,数据的稀疏性的问题日益严重,可扩展性和冷启动的问题也随之而来,协同过滤算法的推荐质量迅速降低。为了进一步提高推荐质量,研究者们对协同过滤算法提出了一系列改进方法,大致可分为两个方向,即基于内存的方法和基于模型的方法。

基于模型的算法主要是利用用户历史数据,采用统计方法或机器学习方法来构建出用户偏好模型,再利用此偏好模型来产生推荐。文献[2]采用贝叶斯网络模型对项目进行预测,采用概率的思想预测用户对项目的评分,取得了不错的效果,但是将贝叶斯网络应用于协同过滤系统,必需要假设各项目的评分独立,而这个假设显然是不成立的,因此当系统项目评分之间的关联性很强的时候,效果往往很差。在基于线性回归的算法中,Slope One是一种线性回归的经典算法,由Daniel Lemire和Anna Maclachlan于2005年提出[3]。Zilei Sun[4]等提出一种改进的Slope One协同过滤系统,Pu Wang[5]等将基于用户的协同过滤和Slope One方法融合,DeJia Zhang[6]等将基于项目的协同过滤和Slope One方法融合,该系列算法简单、高效,且精度与其它复杂度高的算法也不相上下。基于图结构的协同过滤算法将二部图应用到推荐算法中,把用户和产品看成图论的节点,用户与项目之间的评分信息当作连接图中结点的边。周涛和Huang等分别利用用户-产品二部图建立用户-产品关联关系,并据此提出了基于二部图的推荐算法。其中,周涛[7,8]提出了一种基于资源分配的算法,Huang[9]等通过在算法中引入扩散动力学填充评分,缓解数据稀疏性的问题。在矩阵分解模型中,Sawrar等[10]使用SVD分解方法将用户评分矩阵分解为不同的特征及这些特征对应的重要程度,算法利用用户与项目之间潜在

到稿日期:2013-07-19 返修日期:2014-01-06

罗 琦(1958—),男,博士,教授,主要研究方向为动力系统的稳定性、复杂系统分析与综合;缪昕杰(1989—),男,硕士,主要研究方向为数据挖掘、推荐算法、信息检索;魏 倩(1989—),女,硕士,主要研究方向为图像处理、压缩感知。

的关系，对用户-项目评分矩阵的奇异值分解，抽取矩阵中本质的特征，再根据用户对这些特征的喜好程度来预测用户对没有评过分的项目的分数。模型由庞大的历史数据训练而来，其训练代价很大，而一旦用户兴趣改变，模型就必须更新，因此基于模型的协同过滤系统存在不能及时适应用户兴趣变化的缺点。

在基于内存的方法中，最主要的研究方向就是最近邻居的选择问题，准确的最近邻居是提高预测精度的关键。文献[11]提出了一种基于项目的推荐方法，其通过计算项目之间的相似度来产生推荐，在项目和特征相对稳定的系统中，此方法可以达到不错的效果。文献[12]提出了一种在知识层面上比较用户相似度的方法，克服了评分数据极端稀疏带来的影响。文献[13]提出了基于项目聚类的推荐算法，以缩小相似用户的搜索范围，在一定程度上缓解了传统的最近邻搜索存在的问题，从而提高了预测精度。文献[14]提出了一种基于项目分类的推荐算法，以计算同类型用户间的相似度得到目标用户的相似邻居，提高了系统的扩展性和推荐精度。但是这些算法在计算相似性时仍然沿用传统的相似性计算方法，没有考虑相似度计算存在着用户冷启动[15]、偶然性[16]等问题，从而影响了推荐质量。

针对上述问题，在分析用户相似度计算方法的基础上，利用用户间的共同评分数量动态调节用户的相似度，使得目标用户的最近邻更为准确；其次，对于找不到最近邻或者存在冷启动问题的新用户，提出一种新颖的专家信任度排名计算方法，使用专家信任度代替传统的相似度来预测用户的评分。该算法不仅提高了预测的精度，而且弥补了传统推荐算法的不足，有效地缓解了新用户的冷启动问题。

2 协同过滤算法

协同过滤算法可分为以下 3 个步骤：

1)整理数据。获得 $m*n$ 的用户评分矩阵。m 代表用户个数，n 代表项目个数，可以表示为 $U=\{u_1,u_2,\cdots,u_m\}$ 用户集合和 $I=\{i_1,i_2,\cdots,i_n\}$ 项目集合，用户数据采用 $m*n$ 的用户评分矩阵 R_{m*n} 来表示(如表 1 所列)，矩阵的每个元素表示用户对项目的评分，对应着用户对项目的喜爱程度。

表 1 用户-项目评分矩阵 R_{m*n}

	i_1	…	i_j	…	i_n
u_1	R_{11}	…	R_{1j}	…	R_{1n}
…	…	…	…	…	…
u_k	R_{k1}	…	R_{kj}	…	R_{kn}
…	…	…	…	…	…
u_m	R_{m1}	…	R_{mj}	…	R_{mn}

2)K 个最近邻居。通过相似度计算找到评分矩阵中用户之间的相似度，选出与目标用户最相似的 K 个邻居。

3)产生推荐。获得目标用户 K 个最近邻居后，通过预测算法可得到对缺失项目的分数和 Top-N 推荐集。

2.1 相似度计算

传统的相似度计算包括余弦相似性、皮尔森相关性和修正的余弦相似性等。本文主要介绍余弦相似性。

余弦相似度把每个用户看作 V 维项目空间上的一个向量，两个用户之间的相似度则通过向量之间夹角的余弦来判断，夹角越小，相似度越高，值的范围为 0～1。假设用户 u,j 在 V 项目空间上的评分为向量$\vec{u},\vec{j}$，则用户 u 和用户 j 之间的相似度可表示为：

$$\mathrm{sim}(u,j)=\cos(u,j)=\frac{\vec{u}*\vec{j}}{\|\vec{u}\|*\|\vec{j}\|}=\frac{\sum_{i=1}^{n}u_i*j_i}{\sqrt{\sum_{i=1}^{n}(u_i)^2}*\sqrt{\sum_{i=1}^{n}(j_i)^2}} \tag{1}$$

2.2 生成推荐

利用式(1)计算出来的相似度，可按式(2)来预测用户未评分的项目。

$$P_{u,i}=R_u^{avg}+\frac{\sum_{j\in S(i)}\mathrm{sim}(u,j)\times(R_{j,i}-R_j^{avg})}{\sum_{j\in S(i)}(\mathrm{sim}(u,j))} \tag{2}$$

式中，$P_{u,i}$ 表示用户 u 对未评分的项目 i 的预测评分，R_u^{avg}，R_j^{avg} 分别表示用户 u 和用户 j 的平均评分，$\mathrm{sim}(u,j)$ 为用户 u 和用户 j 的相似度，$S(i)$ 为用户 u 的最近邻集合。

2.3 算法分析

在数据非常稀疏的矩阵中，使用传统的协同过滤推荐算法存在几点不足：

1)相似度度量的标准应该是两两用户间共同评分的数目越多，对同一个项目的评分越接近，相似度则越高。但是如果当用户间只存在一个或很少的共同评分项目时，直接使用余弦相似度来计算，他们之间的相似度会非常高。例如表 2 中与用户 A 相似度最高的应该是用户 C，他们拥有 3 个共同评分项目，分数也比较接近，但用余弦相似度求出的 A 和 B 的相似度却超过用户 A 和 C 之间的相似度，这显然不符合我们的要求。而在实际的数据集中，用户间的共同评分往往很多都是 1 个或很少，这使得我们在最近邻的选择和预测分数上都存在许多偏差。

表 2 数据集

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
A	5	3	3	3			1	5		
B	5								2	
C			5			5	5	5	4	
D					5					4
E										

2)用户与其他用户之间的相似度为零或者他是刚加入系统的新用户，传统的推荐系统也是无法为其推荐和预测分数的。例如 E 用户为新注册的用户，他没有对任何项目评过分，传统的相似度则算不出其最近邻，这就是我们所说的用户冷启动问题；还有用户 D 虽然评过两次分，但他并没有与任何用户有共同评分项目，传统方法也无法计算他的最近邻。

3)忽略了用户之间的信任关系，在实际生活中，用户之间存在不同程度的信任度，除了相似度以外，信任度也会导致不同的推荐结果。

3 改进算法

3.1 算法思路

针对 2.3 节中 1)的不足，本文参照用户间共同评分项目个数越多，则他们的兴趣偏好越相似的原则，利用不同用户间的共同评分项目个数来调整相似度。我们选择了几种符合此种原则的函数进行对比分析，包括二次函数、三次函数、根函数和反正切函数等。通过计算发现反正切函数更符合我们的需求，并且与文献[20]中高斯加权以及文献[21]指数函数进行了对比，精度也优于这两种方法；故利用反正切函数对原相

似度赋权，动态调整共同评分次数对原相似度的影响，以得到更准确的最近邻，提高协同过滤算法的精度。

针对 2.3 节中 2)和 3)的不足，根据社会网络中 FTL 模型(follow the leader)[17]的思想，即专家对大众用户的抉择影响，在传统的协同过滤算法的基础上，计算每个用户的影响力，用户的影响力越大，他的评分越有参考价值，利用这种影响力来代替传统的相似度，以预测新用户或找不到最近邻的用户的评分，弥补传统算法的不足，缓解新用户的冷启动问题。

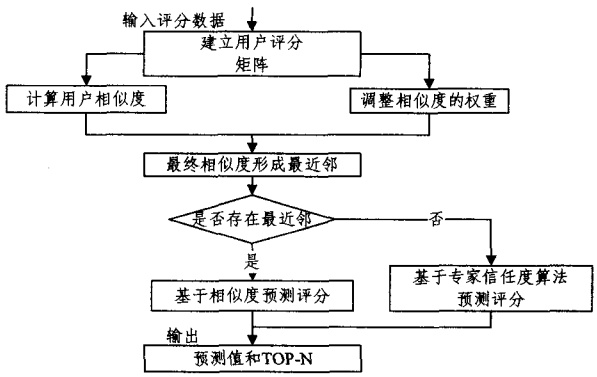

图 1　算法工作流程图

3.2　改进相似度的预测评分算法

在实际推荐系统中，由于评分数量非常稀疏，共同评分数目极少，因此在计算用户间相似度时，推荐系统使用传统相似度方法会出现大量类似算法分析中 1)提到的问题，使得我们最后找到的邻居并不是目标用户准确的最近邻，预测分数和推荐也会受到影响。为了提高相似度算法的精确度，不少研究者对最近邻的选择进行了研究。Herlocker 等在文献[18]中，使用两两用户共同评分项目的个数除以设定好的阈值得到一个相似度约束系数，并使用该系数对用户间的评分相似度进行修正。然而该算法需要预先定义一个阈值，该阈值必须通过交叉验证才能得到，这将导致计算更加复杂。

$$\mathrm{sim}'(u,v)=\frac{\min(|I_u\cap I_v|,\gamma)}{\gamma}*\mathrm{sim}(u,v)$$

式中，$I(u)\cap I(v)$为两个用户的共同评分个数，γ为设定的阈值。

本文参照用户间共同评分项目个数越多，则他们的兴趣越相似的原则，利用不同用户间共同评分项目个数来调整相似度。

改进算法采用反正切函数调整共同评分次数对相似度的影响，改进的相似度算法如下：

$$U\mathrm{sim}(u,v)=\arctan\left(\frac{I(u)\cap I(v)}{\max|I(u)|\cap|I(w)|}\right)*\mathrm{sim}(u,v) \tag{3}$$

用户 u 和 v 的共同评分的项目用 $I(u)\cap I(v)$表示，$\max|I(u)|\cap|I(w)|$表示用户 u 与其他用户中共同评分项目个数的最大值。从式(3)可看出调整系数反映在用户 u 与用户 v 所关注的项目中，重合的项目越多，则 u 与 v 的相似度越大。

从表 2 的数据中可以看出与用户 A 相似度最高的应该是用户 C，他们具有 3 个共同评分项目，且评分也相对接近；然而从表 3 看出，使用传统相似度计算用户 A 与用户 B 的相似度为 0.52；用户 A 与用户 C 的相似度为 0.47，这显然是不正确的。通过我们改进的算法计算得出用户 C 变成了用户 A 最相似的邻居，达到了我们预期的效果。

表 3　相似度对比

传统相似度					
	A	B	C	D	E
A	1	0.52	0.47	0	0
B	0.52	1	0.13	0	0
C	0.47	0.13	1	0	0
D	0	0	0	1	0
E	0	0	0	0	1
改进相似度					
	A	B	C	D	E
A	1	0.16	0.37	0	0
B	0.41	1	1	0.10	0
C	0.37	0.04	1	0	0
D	0	0	0	1	0
E	0	0	0	0	1

算法 1　改进相似度的协同过滤算法

输入：用户数量 m，项目数量 n，用户-评分项目矩阵 R_{m*n}

输出：预测评分 $P_{u,i}$。

a)采用式(3)计算用户间的相似度，建立用户相似度矩阵。

b)在相似度矩阵中查找目标用户 u 的最近邻集合 $S_j=\{I_1,I_2,\cdots,I_n\}$，且 I_1 与用户 u 的相似度最高，I_2 用户次之，以此类推。

c)最后采用式(2)预测用户未评分的项目。

3.3　专家信任度

从传统的预测评分公式(2)可以看出，必须要有两个必要条件：1)用户间存在相似度；2)最近邻对预测项目有过评分。在极度稀疏的数据中，用户间的相似度很多都是零，例如表 3 中的数据，用户 D 和用户 E 在相似度矩阵中相似度都为零，即得不到他们的最近邻居，许多的评分用传统方法无法预测，这说明单纯使用相似度推荐，并不能够满足我们的需要。

根据上述的分析，联想到文献[17]中提出的 follow the leader 模型，该模型将社会网络分为两部分，包括普通人和专家，其认为普通人往往更倾向于专家的意见，专家的影响力越高，常常对普通人的判断产生越重要的影响。本文提出专家信任度的定义，通过计算出每个用户的专家信任度，使用专家信任度代替传统的相似度来预测用户的评分。

在文献[22]中，作者分析用户活跃度和物品流行度的关系时指出：一般新用户或者不活跃的用户往往倾向于浏览热门的物品，因为他们对网站还不熟悉，只能点击首页的热门产品。为了使不活跃的用户或新用户得到更好的推荐，本文对用户的专家信任度的定义标准包括：

1)用户的评分数量越多，活跃度越高，专家信任度越高；

2)用户的评分越精确，质量越高，专家信任度越高；

3)在前两个定义的基础上，用户选择的热门物品越多，信任度越高。

本文根据上述定义的思想，通过参考 HITS 网页权威度算法，提出专家信任度算法。设用户数目为 m，物品数目为 n，两个向量 E 和 Q 分别表示用户专家信任度和物品的流行度分数，向量初始化都为 1。那么关于专家分数向量和物品分数的向量的关系式可以表示为：

$$\begin{cases}E_{n+1}=aA^{\mathrm{T}}Q_n\\Q_{n+1}=\beta AE_n\end{cases} \tag{4}$$

式中，a,β为归一化系数，分别定义为 E 和 Q 中的最大值，评分质量体现在邻接矩阵 A 中，用户对项目有过评分，则用式(5)计算，反之，则为零。

$$A_{u,i}=1-\frac{|r_u^i-R_{avg}^i|}{r_{\max}^i} \tag{5}$$

式中，r_u^i 表示用户 u 对项目 i 的评分，R_{avg}^i 为项目 i 得到的平均分，$r_{\max}^i$ 为项目 i 得到的最大分值，那么 $A_{u,i}$ 则可以表示用户 u 对项目 i 的评分与项目 i 得到的平均分的偏离程度。偏离程度越大，则 $A_{u,i}$ 越小；反之，$A_{u,i}$ 越大。按式(4)迭代计算 K 次，直至收敛，即用户专家排名和物品分数排名不再发生改变为止。

依然使用表 2 中的数据计算用户的专家信任度排名，从表 4 中的专家信任度可以看出，首先排名高的用户必须是活跃用户，即拥有大量的评分次数，比如用户 A。其次，选择的物品大多都是热门物品，比如用户 B 和用户 D 都选择了两个物品，然而用户 B 选择的物品比较热门，用户 D 选择的比较冷门，所以用户 B 的排名要比用户 D 的高，最后通过改进邻接矩阵 A，使得用户评分偏离项目平均值的程度越小，用户的专家信任度越高。

表 4　专家信任度排名

	专家信任度 E	排名 L
A	1	1
B	0.28256	3
C	0.71711	2
D	0.10796	4
E	0	5

算法 2　专家信任度算法

输入：用户数量 m，项目数量 n，评分数据 R_{m*n}，迭代次数 K

输出：用户专家信任度 E 和信任度排名列表 L

设置 E 向量为 $(1,1,\cdots,1)\in Q^m$

设置 Q 向量为 $(1,1,\cdots,1)\in Q^n$

A←生成邻接矩阵(R，$A_{u,i}$)

for i=1 to K do

　E←$Q*A^T$

　Q←$E*A$

　E←E/max(E)

　Q←Q/max(Q)

endfor

L←通过 E 对用户排序生成排名列表

return L，E

3.4　基于专家信任度的预测评分算法

在计算基于专家信任度的预测评分算法时，选取专家信任度最高的 J 个专家作为最近邻。通过实验验证，一般专家数量 J 取为用户数量的平方根最为合适。这里选取用户 a 对项目 i 的预测评分。

$$P_{a,i}=R_a^{avg}+\frac{\sum_{u=1}^{J}E(u)*(r_u^i-R_u^{avg})}{\sum_{u=1}^{J}E(u)} \tag{6}$$

式中，$P_{a,i}$ 为用户 a 对项目 i 的预测评分，J 为对项目 i 评过分的专家数量，r_u^i 为专家 u 对项目 i 的实际评分。R_a^{avg} 和 R_u^{avg} 为用户 a 和专家 u 的平均评分。

对于刚注册的新用户，我们直接使用专家信任度最高的用户作为其最近邻，预测新用户的评分，在式(6)的基础上，采用式(7)来缓解冷启动的问题。

$$P_{a,i}=\frac{\sum_{u=1}^{J}E(u)*r_u^i}{\sum_{u=1}^{J}E(u)} \tag{7}$$

4　实验分析

4.1　数据集

本实验采用的数据来源于 Movie lens 站点提供的 m1 公开数据集(http://movilens.umn.edu/)。它是由 943 个用户和 1682 个项目组成的 100000 条评分数据，每个用户对项目都必须至少有 20 条以上的评价。用户评分范围为 1～5，分值越大表示用户对项目的喜爱程度越大。实验采用的是已经划分好的 80%的训练集和 20%的测试集，评分数据集的稀疏度定义为 1－已评分项目/总评分项目，因此本实验数据集的稀疏度为：

$$1-\frac{100000}{943*1682}=93.69\%$$

4.2　度量标准

采用平均绝对误差(MAE)度量推荐系统的推荐质量。MAE[18]通过计算预测项目集合与实际项目评分集合的偏差来判定预测的准确性，MAE 值越小，推荐精度则越好。

设预测用户的评分集合表示为 $\{p_1,p_2,\cdots,p_N\}$，实际用户评分集合表示为 $\{q_1,q_2,\cdots,q_N\}$。则

$$MAE=\frac{\sum_{i=1}^{N}(p_i-q_i)}{N} \tag{8}$$

4.3　改进相似度算法的比较

将改进相似度算法(算法 1)与传统的协同过滤算法(User-Base)、文献[19]中的高斯加权、文献[20]指数函数进行 MAE 值对比，实验的目的是体现出我们改进的算法优于其他改进的算法。横坐标选为最近邻居个数 K，纵坐标为 MAE 值，邻居个数从 10 开始，增至 50 个。

如图 2 所示，所有的算法随着最近邻居的个数的增加，MAE 值都在降低，最终稳定在一个位置。改进的协同过滤算法精度相对于传统的协同过滤和其他改进算法都有一定的提高，并且随着邻居个数的增加，MAE 值减小，在最近邻居个数为 50 时，MAE 值基本稳定，达到 0.7221。实验证明采用反正切函数改进相似度计算是可行的。

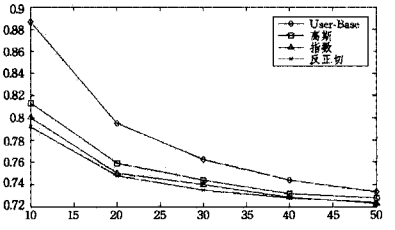

图 2　比较相似度算法 MAE

4.4　相似度分布

为了更具体地体现出随着数据集的减少，部分用户会出现找不到最近邻的情况，实验引入变量 x 来表示训练集占数据集的百分比。实验中将 x 分成 30%到 80%，分别考察不同大小的训练集中相似度的分布情况。横坐标表示相似度的大小，纵坐标表示相似度在某个范围出现的次数。

如图 3 所示，当训练集的数据变少以后，相似度为零的数量不断地增加，这说明稀疏的数据集直接导致用户间的共同评分数目变得更少，越来越多的用户找不到最近邻，预测精度也会变得很差。在这种情况下，推荐系统则有必要找到一种可以替代相似度的算法，来为找不到最近邻的用户预测分数或作出推荐。

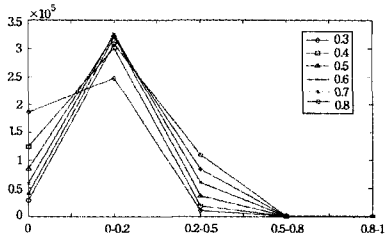

图 3　相似度分布

4.5　不同比例的 x 中精度对比

实验数据集中训练集/数据集的比率分成 0.3～0.8，将引入专家信任度的算法与改进相似度算法(Improve CF)和传统的协同过滤算法(KNN)作比较，在计算 MAE 值时，最近邻和专家个数都选择为 50。实验的目的是证明在引入专家信任度算法后，在不同的 x 下，算法都能有比较好的精度值。

4.6　相同比例的 x 中精度对比

在训练集/数据集为 80%/20%的数据集中，同样将引入专家信任度的算法与改进相似度算法(Improve CF)和传统的协同过滤算法作比较，在计算 MAE 值时，邻居个数或专家个数从 10 个增至 50 个。实验的目的是为了证明在引入专家信任度算法后，在相同的训练集、不同的最近邻居个数下，算法都能有比较好的精度值。

从图 4 和图 5 的实验中可以看出，在同等数据集及相同或不同的 x 比例的基础上，引入专家信任度的算法都要比传统的协同过滤算法精度高很多。实验证明，将专家信任度算法引入到协同过滤算法中是可行的。

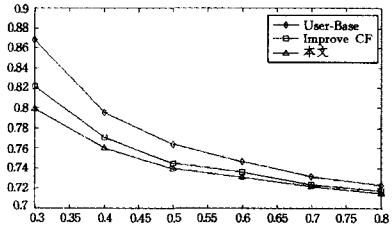

图 4　不同数据集中 MAE 比较

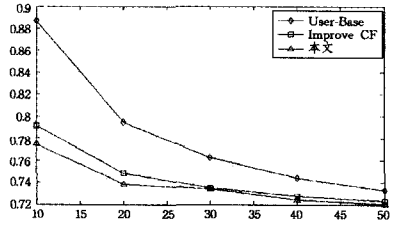

图 5　MAE 比较

结束语　本文构建了切实可行的协同过滤算法，改进了传统方法过度依赖相似度的缺点，通过考虑专家对大众的抉择影响，提出了基于专家信任模型的推荐算法。通过实验表明，专家信任度是影响推荐的重要因素之一，该算法的精度质量优于传统的协同过滤算法，并弥补了传统算法中一直存在的用户冷启动问题。今后的研究方向是进一步改进专家信任度的算法，包括加入对用户信息和项目分类信息的处理，同时项目冷启动[23]也是未来需要研究的问题。

参 考 文 献

[1] Koren Y, Bell R. Advances in. Collaborative. Filtering[EB/OL]. http://research. yahoo. com/pub/3503, 2011-09-26

[2] Chen Y H, George E I. A Bayesian Model for Collaborative Filtering[C]// Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics. 1999: 56-60

[3] Lemire D, Maclachlan A. Slope One Predictors for Online Rating Based Collaborative Filtering [C]// SIAM Data Mining(SDM'05). 2005: 21-23

[4] Sun Zi-lei, Luo Nian-long, Kuang Wei. One Real-Time Personalized Recommendation Systems based on Slope One Algorithm [C] // 8th International Conference on Fuzzy Systems and Knowledge Discovery. 2011: 1826-1830

[5] Wang Pu, Ye Hong-wu. A Personalized Recommendation Algorithm Combining Slope One Scheme and User Based Collaborative Filtering [C]// International Conference on Industrial and Information Systems. 2009: 152-154

[6] Zhang De-jia. An Item-based Collaborative Filtering Recommendation Algorithm Using Slope One Scheme Smoothing [C]// 2nd International Symposium on Electronic Commerce and Security. 2009: 215-217

[7] Zhou T, Ren J, Medo M, et al. Bipartite Network Projection and Personal Recommendation[J]. Phys Rev E, 2007, 76: 046115

[8] Zhou T, Jiang L L, Su R Q, et al. Effect of Initial Configuration on Network Based Recommendation[J]. Europhys Lett, 2008, 81: 58004

[9] Huang Z, Chen H, Zeng D. Applying as Sociative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering [J]. IEEE Trans Information Systems, 2004, 22(1): 116-142

[10] Sarwar B M, Karypis G, Konstan J A, et al. Application of Dimensionality Reduction in Recommender System-A Case Study [C]// ACM 2000 KDD Workshop on Web Mining fore-commerce-Challenges and Opportunities. Boston, MA, 2000

[11] Sarwar B, Karypis G, Konstan J, et al. Item-Base collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International World Wide Web Conference. 2001: 285-295

[12] 张光卫，李德毅，李鹏，等. 基于云模型的协同过滤推荐算法[J]. 软件学报，2007，18(10)：2403-2411

[13] 邓爱林，左子叶，朱扬勇. 基于项目聚类的协同过滤推荐算法[J]. 小型微型计算机系统，2004，25(9)：1665-1670

[14] 熊忠阳，刘芹，张玉芳，等. 基于项目分类的协同过滤改进算法[J]. 计算机应用研究，2012，29(2)：493-496

[15] Sarwar B M, Karypis G, Konstan J A, et al. Application of dimensionality reduction in recommender system-a case study[C]// Proc. of the ACM WebKDD 2000 Workshop. 2000

[16] 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)：1369-1376

[17] Goldbaum D. Follow the leader: Simulations on a dynamic social-network[EB/OL]. http://www. business. uts. edu. au/finance/research/wpapers/wp155. pdf, 2011-09-26

[18] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating Collaborative Filtering Recommender Systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 5-53

[19] Herlocker J, Konstan J, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms [J]. Information Retrieval, 2002, 5(4): 287-310

[20] 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

[21] 陶维安，范会联. 基于评分支持度的最近邻协同过滤推荐算法[J]. 计算机应用研究，2012，29(5)：1723-1725，1728

[22] 项亮. 推荐系统实现[M]. 北京：人民邮电出版社，2012

[23] Zhang Zi-ke, Liu Chuang, Zhang Yi-cheng, et al. Solving the Cold-Start Problem in Recommender Systems with Social Tags [J]. EPL(Europhysics Letters), 2010, 92(2): 28002