一种基于 Sigmoid 函数的改进协同过滤推荐算法*

方耀宁，郭云飞，扈红超，兰巨龙
（国家数字交换系统工程技术研究中心，郑州 450002）

摘　要： 随着电子商务和社交网络的蓬勃发展，推荐系统逐渐成为数据挖掘领域的重要研究方向。推荐系统能够从海量信息中定位用户兴趣点，提供个性化服务。协同过滤算法能够有效分析用户偏好，提供合适的推荐服务。针对评分矩阵稀疏时传统协同过滤算法性能很差的问题，提出一种基于 Sigmoid 函数的改进推荐系统算法。利用 Sigmoid 函数对不同项目进行建模，得到项目的平均受欢迎程度；利用 Sigmoid 函数对不同用户进行建模，将评分映射为用户对项目的喜好程度；根据用户对项目喜好程度应该与项目平均受欢迎程度贴近的原则进行评分预测。在两组真实数据集合上的实验结果表明，该算法较好地解决了数据稀疏性问题，能够有效提高传统算法的预测准确性。

关键词： 推荐系统；协同过滤；稀疏性问题；Sigmoid 函数
中图分类号： TP18；TP301.6　　**文献标志码：** A　　**文章编号：** 1001-3695(2013)06-1688-04
doi：10.3969/j.issn.1001-3695.2013.06.022

Improved collaborative filtering recommender algorithm based on Sigmoid function

FANG Yao-ning, GUO Yun-fei, HU Hong-chao, LAN Ju-long
(*National Digital Switching System Engineering & Technological R&D Center*, *Zhengzhou* 450002, *China*)

Abstract: As the rapid development of the electronic commerce and social network, recommender systems have become one of the most important research areas in data mining field. Recommender systems can identify users' interest out of humorous information in order to provide personalized service. Collaborative filtering (CF) is efficient in extracting users' preferences and making proper recommendations. To address the data sparsity problem of classic CF algorithms and improve the performance, this paper introduced an improved algorithm based on Sigmoid function. Different items were modeled with Sigmoid function in order to capture their popularity, while different users were modeled to map ratings into preferences. Predictions were made according to that preferences should keep consistent with popularities. Experimental results on two real world datasets show the proposed method can alleviate the sparsity problem and are effective to improve the performance of classic CF algorithms.
Key words: recommender systems; collaborative filtering(CF); sparsity problem; Sigmoid function

0　引言

互联网的飞速发展将人类带入了信息膨胀的时代，从海量信息中获取所需内容却成为了一件复杂的事情——信息过载(information overload)。与传统的搜索引擎不同，推荐系统利用数据挖掘、人工智能等技术能够主动帮助人们过滤信息，提供个性化的服务。在过去的十多年里，商业利益驱动的电子商务和用户体验驱动的社交网络，极大地促进了推荐系统的研究与发展。推荐系统不仅可以提供娱乐、生活方面的服务推荐，如电影推荐、好友推荐，而且可以提供更加专业化的推荐，如科技论文推荐、专利技术推荐[1~3]。

推荐系统主要分为基于内容的(content-based)推荐系统、协同过滤(collaborative filtering)推荐系统和混合(hybrid approaches)推荐系统三大类，其他还有基于谱分析、扩散的推荐方法等[1,2]。基于内容的协同过滤推荐系统侧重用户(user)和项目(item)的内容信息分析，如年龄、性别、教育背景等个人信息会暗示用户的兴趣，色彩、种类、价格等信息会暗示商品面向的消费群体；协同过滤算法从用户以往的行为信息中挖掘用户的兴趣和偏好，侧重的是用户与项目之间的二元关系。协同过滤推荐系统结构简单，不需要具备特定领域内的知识，能处理不易进行内容分析的艺术品、音乐等，是目前最为成功的推荐系统[1,3]。

协同过滤基于如下假设：相似用户对同一项目的评分是相似的，相似项目被同一用户的评分也是相似的，这是一种“物以类聚，人以群分”的思想[4]。现实中的评分数据是非常稀疏的，MovieLens 数据集的密度是 4.5%，Netflix 是 1.2%，这些是为了进行实验而挑选的稠密评分数据集合；BookCrossing 的密度为 0.0015%，Bibsonomy 是 0.35%，Delicious 是 0.046%。协同过滤的原理决定了在评分矩阵非常稀疏的情况下算法性能很差，即稀疏问题[1,4]。为了解决评分矩阵的稀疏问题，提高协同过滤算法性能，本文提出了 iSensity 协同过滤推荐方法，即在评分矩阵稀疏时，利用 Sigmoid 函数融合全局和局部信息进行决策。

收稿日期： 2012-09-11；**修回日期：** 2012-10-26　　**基金项目：** 国家“973”计划资助项目(2012CB315901)；国家“863”计划资助项目(2011AA01A103)

作者简介： 方耀宁(1987-)，男，山东寿光人，硕士研究生，主要研究方向为复杂网络、数据挖掘(fxn07@163.com)；郭云飞(1963-)，男，教授，博导，主要研究方向为高性能交换技术、下一代互联网；扈红超(1982-)，男，博士，主要研究方向为高性能路由器、网络流量均衡技术和业务管控技术；兰巨龙(1962-)，男，河北张北人，教授，博导，主要研究方向为宽带信息网络、高速路由器核心技术.

1　相关研究

协同过滤算法主要分为 model-based 和 memory-based 两类。Model-based 算法首先学习一种模型,然后通过模型来进行推荐,如基于矩阵分解的算法、基于网络的算法;memory-based 算法基本思想是寻找相似的用户/项目,根据相似的用户/项目进行预测[3,4]。

基于用户/项目相似性的 K-近邻算法(K-nearest neighbor, K-NN)实际上是一种局部插值算法,忽略了全局信息,在评分矩阵稀疏的情况下误差较大。基于用户和项目的算法是应用最早的协同过滤推荐算法,这两种经典算法原理简单,在评分矩阵密度高时性能良好,但两者都需要消耗较高的内存和计算时间[5]。Lemire 等人[6]提出的 SlopeOne 算法,原理简单易于实现,内存消耗较小,计算时间短,但对评分矩阵的密度和尺寸依赖性较强。

基于矩阵分解的算法如 RSVD(regular singular value decomposition)、RSVD2、SVD ++[3,5,7],可以起到降低维度的作用,具有较好的推荐性能,但是需要调节的参数较多,计算时间非常长。

基于隐含特征提取的推荐算法还有概率潜在语义分析(latent semantic analysis, pLSA)[8]、潜在狄利克雷分配模型(latent Dirichlet allocation, LDA)[9]、受限玻尔兹曼机(restricted Boltzmann machines, RBM)[10]等。基于社会网络和扩散的推荐算法[2]也成为最近研究的一个热点。这些方法都是从其他研究领域移植到协同过滤中的,在实际推荐系统中的高效性还有待进一步的研究。

文献[5,11,12]在真实世界数据集合上对各种协同过滤算法进行了对比实验并作了比较,从而发现不同算法的性能对稀疏度的依赖程度不同;推荐算法在预测准确性、覆盖率、计算时间、内存消耗上存在着复杂的关系。基于矩阵分解的算法虽然精确度高,但是计算量非常大,很难应用在实际项目中;K-NN 和 SlopeOne 算法可行性强,应用最广,但是在矩阵密度低时性能非常差。

评分矩阵稀疏是推荐系统的固有特征,而协同过滤依赖的恰恰是用户的历史行为信息。以电影推荐系统为例,网站上存储着成百上千万部的电影,而用户进行评分过的电影不过几百部甚至几部,这样用户—项目评分矩阵就会非常稀疏,存在大量的零元素,因此就很难进行相似性或者潜在因子的计算,从而影响算法性能。

通过对用户/项目进行聚类,用分类内部的平均值来填充缺失的评分,从而使得评分矩阵变密集[13]。实际上填充随机值或项目均值也能改善数据稀疏性问题。由于现实中的商品一般都有各自的种类,不需要再进行聚类,这样可以部分解决数据稀疏性和冷启动的问题,但是用聚类来降低分辨率的方法不仅复杂度高,而且性能也不理想[5]。

Huang 等人[14]在协同过滤算法中引入了扩散动力学,通过相似性的扩散可以找到更多相似的邻居(链路预测)[15]。Yildirim 等人[16]用随机游走的算法预测项目之间的关系来进行推荐,从而解决稀疏性问题。Esslimani 等人[17]用链路预测的方法实现了一种致密化行为网络协同过滤模型(densified behavioral network based collaborative filtering, D-BNCF)。这类链路预测方法是建立在相似性可以传播的基础上的,而且计算量非常大。

近年来,Liu 等人[18]提出了 iExpand 推荐系统,通过挖掘用户潜在兴趣点的转移来提高推荐准确性,一定程度上解决了评分矩阵的稀疏性问题,但是需要调节非常多的参数。Yang 等人[19]提出了一种基于社交网络的贝叶斯推荐系统,改进了经典的概率协同过滤算法,能够较好地处理评分矩阵稀疏和冷启动的问题,但是需要合理设定先验分布(prior distribution)的参数。

目前,还没有一种通用的方法来有效解决评分矩阵稀疏性的问题。

2　基准算法

2.1　K-近邻算法

2.1.1　相似性度量

相似度的计算在 K-NN 算法中是首要的步骤,目前主要有三种度量用户间相似性的方法,即余弦相似性、修正的余弦相似性和相关相似性。同时,改进的相似性计算方法不断被提出[5]。

a)余弦相似性度量方法是把用户对项目评分看做是 n 维空间上的向量,对于没有评分的项目将评分值设为默认值(如零、均值或其他填充方法),然后通过计算向量间的余弦夹角来度量用户间的相似性。

$$s_{ij}=\cos(r_i,r_j)=\frac{r_i\cdot r_j}{\|r_i\|\times\|r_j\|}=\frac{\sum_{v\in V_{ij}}r_{iv}r_{jv}}{\sqrt{\sum_{v\in V_i}r_{iv}^2}\sqrt{\sum_{v\in V_j}r_{jv}^2}}$$

其中:V_i 和 V_j 分别表示用户 i、j 评分过的项目集合;V_{ij} 表示 i、j 共同评分过的项目集合。

b)修正的余弦相似性考虑到用户评分尺度问题,在数据预处理阶段将所有用户对项目的评分减去了相应用户的平均评分。

$$s_{ij}=\frac{\sum_{v\in V_{ij}}(r_{iv}-\overline{r_i})(r_{jv}-\overline{r_j})}{\sqrt{\sum_{v\in V_i}(r_{iv}-\overline{r_i})^2}\sqrt{\sum_{v\in V_j}(r_{jv}-\overline{r_j})^2}}$$

c)皮尔森(Pearson)相关系数是一种常用的相关性计算方法。相关性与相似性有着很大的联系,可以用相关系数来刻画相似性程度。用户 i 和 j 的相关系数 p_{ij} 和相似性 s_{ij} 为

$$s_{ij}=\frac{np_{ij}}{n+\lambda},p_{ij}=\frac{\sum_{v\in V_{ij}}(r_{iv}-\overline{r_i})(r_{jv}-\overline{r_j})}{\sqrt{\sum_{v\in V(i,j)}(r_{iv}-\overline{r_i})^2}\sqrt{\sum_{v\in V(i,j)}(r_{jv}-\overline{r_j})^2}}$$

GroupLens 最初用 p_{ij} 直接作为用户间的相似性,后来加入 $\frac{n}{n+\lambda}$ 项保证了两个用户共同评分的项目越多,则相似性越强,其中的原理可以从贝叶斯分析来解释[3]。λ 取值根据数据集合的特点来确定。

2.1.2　基于用户的 K-NN 算法

数据预处理阶段需要计算不同用户之间的相似性,需要耗费较多的内存和计算时间,但可以离线进行。为了消除不同用户评分尺度的差异,在计算过程中评分要减去用户对项目的平均评分[10]。从数据集合中找出与目标用户 u 最相似且对项目 i 评分过的 K 个用户作为邻居集合 U_{ui},然后把 K 个邻居对项目 i 评分的加权平均值作为用户 u 对项目 i 的评分预测值。

$$\hat{r}_{ui}=\overline{r_u}+\frac{1}{\sum_{v\in U_{ui}}|s_{uv}|}\sum_{v\in U_{ui}}s_{uv}(r_{vi}-\overline{r_v})$$

其中:s_{uv} 表示用户 u 和 v 的相似性;$\overline{r_u}$、$\overline{r_v}$ 分别表示用户 u、v 的评分均值。

2.1.3 基于项目的 K-NN 算法

数据预处理阶段计算的是项目与项目之间的相似性,原理与基于用户的一致,只是不太符合人们的思维习惯。研究表明,基于项目相似性的 K-近邻算法比基于用户的性能更好[12],其原因可能是项目之间的相似性比用户之间的相似性更加稳定[4]。

$$\hat{r}_{ui}=\frac{1}{\sum_{j\in V_{ui}}|s_{ij}|}\sum_{j\in V_{ui}}s_{ij}r_{uj}$$

其中:s_{ij}表示项目 i 和 j 的相似性;V_{ui}表示用户 u 已经评分的项目中与项目 i 最相似的 K 个项目的集合。

2.2 SlopeOne

SlopeOne 算法[6]的基本原理是假设同一个用户对不同的项目评分之间存在线性关系,并用项目之间的平均差来表示这种关系。图 1 例子中,用户 b 对项目 j 的预测评分为:$x=3+(2.5-2)=3.5$。

	item *i*		item *j*
user *a*	2		2.5
user *b*	3		*x*

图1 SlopeOne例子

数据预处理阶段计算不同项目之间的平均差 s_{ij},然后根据目标用户 u 已经选择的项目集合 V_u 预测项目 i 的评分$\hat{r}_{ui}$。

$$\hat{r}_{ui}=\frac{\sum_{j\in V_u}|U_{ij}|(r_{uj}+s_{ij})}{\sum_{j\in V_u}|U_{ij}|},s_{ij}=\frac{\sum_{u\in U_{ij}}r_{ui}-r_{uj}}{|U_{ij}|}$$

其中:U_{ij}表示共同评价过项目 i 和 j 的用户集合,$V(u)$表示用户 u 评价过的项目集合。虽然 SlopeOne 结构形式很简单,但有时比基于用户和项目的 K-NN 算法性能要好[5,12]。

3 iSensity 协同过滤推荐算法

Sigmoid 函数是一种常用的阈值函数,具有很多良好的性质:光滑、单调性、对称性等。很多自然过程包括复杂系统的学习曲线,都呈现出 Sigmoid 函数的特征:较小的初始值,加速增长,加速度逐渐减小,最后逐渐稳定。

$$f(x)=\frac{1-e^{-\beta x}}{1+e^{-\beta x}}=$$

$$\frac{2}{1+e^{-\beta x}}-1\quad\beta>0$$

Sigmoid 函数曲线非常符合人类的情感分布:横轴表示用户评分,纵轴表示情感强度;$f(x)$为正时表示用户喜欢该项目,反之则表示不喜欢;$f(x)$越接近 1,表示用户越喜欢该项目,$f(x)$越接近 -1,表示用户越厌恶该项目;在原点处情感非常敏感,$f(x)$变化较快,在远离原点的地方情感逐渐饱和麻木,$f(x)$变化较缓。

本文提出的 iSensity 算法利用 Sigmoid 函数的特性,分别对项目和用户进行建模,表征出项目的受欢迎程度和用户对项目的喜爱程度,消除了不同用户评分尺度的差别。在建模过程中充分利用了全局信息,从而克服了评分矩阵稀疏性问题,提高了推荐系统预测准确性。

3.1 数据预处理

数据预处理步骤如下:

a)计算每个分类内项目的平均值,CM_i 表示项目 i 所在分类的平均评分。

b)计算每个待推荐项目的平均值,IM_i 表示项目 i 的平均评分。

c)计算每个用户不同分类内的平均值,UM_{ui}表示用户 u 对项目 i 所在分类的平均评分。

3.2 iSensity 算法

iSensity 算法步骤如下所示,其中[ ]为高斯符号。iSensity 算法计算复杂度低,速度快,容易与其他算法进行融合。如果把$\hat{r}_{ui}=UM_{ui}+f_{Uu}^{-1}f_{Ii}(IM_i-CM_i)$中得到的$\hat{r}_{ui}$替换为 UCF 或 SlopeOne 算法计算得到的与测评分,就实现了利用 Sigmoid 函数对 UCF 或 SlopeOne 算法的改进。

算法 1 iSensity

Input: Training set S, test set T, user number M, item number N

for u = 1,2,…,M do

$f_{Uu}(x)=2(1+e^{\frac{-2x}{\max R-UM_{ui}}})^{-1}-1$

end

for i = 1,2,…,N do

$f_{Ii}(x)=2(1+e^{\frac{-2x}{\max R-CM_i}})^{-1}-1$

end

for (u,i) ∈ T do

$f_{Ii}(IM_i-CM_i)=f_{Uu}(\hat{r}_{ui}-UM_{ui})$

$\Rightarrow\hat{r}_{ui}=UM_{ui}+f_{Uu}^{-1}f_{Ii}(IM_i-CM_i)$

if $f_{Uu}([\hat{r}_{ui}]+1-UM_{ui})-f_{Ii}(IM_i-CM_i)\leqslant f_{Ii}(IM_i-CM_i)-f_{Uu}([\hat{r}_{ui}]-UM_{ui})$

then $\hat{r}_{ui}=[\hat{r}_{ui}]+1$

else $\hat{r}_{ui}=[\hat{r}_{ui}]$

end if

end

4 实验设计及结果分析

4.1 实验数据集合

本文选取两组真实世界的测试数据,即 MovieLens 和 Book-Crossing。MovieLens 是由 GroupLens Research Project 收集的电影评分信息,是目前最常用的评估推荐系统的数据集之一。MovieLens100 K 的数据集包含 943 个用户、1 682 个项目的 10 万条评分信息;Book-Crossing 数据集包含了用户对书籍的真实评分。本文随机抽取了部分 Book-Crossing 的数据作为数据集合[14],如表 1 所示。

表 1 数据集合信息

name	domain	user	item	records	density/%
MovieLens	Movie	943	1 682	100 000	6.3
Book-Crossing	Book	868	1 636	17 213	1.2

4.2 实验设计

从 MovieLens 数据集合随机抽取了 $x\%$($x=10$、15、20%、25、30、35)作为训练集合,其余$(100-x)\%$作为测试集合;从 Book-Crossing 数据集合中抽取了 $x\%$($x=40$、50、60、70、80、90)作为训练集合,其余$(100-x)\%$作为测试集合。每组实验随机生成测试集合训练集,重复 10 次,实验结果取平均值。

Book-Crossing 的项目数据没有分类信息,可以用全局平均值代替分类内项目的平均值,用户平均值代替用户不同分类内的平均值。

对比了 iSensity、ItemAvg、UCF、SlopeOne 的性能,并用 Sigmoid 函数对后两种方法进行了改进(UCF +、SlopeOne +)。

UCF 采用了皮尔森相似度公式,寻找相似度为正的用户

作为邻居。考虑到评分均值往往不在评分范围的中间位置,将 Sigmoid 函数设计为不对称形式:

$$f_{Ii}(x)=\begin{cases}2(1+e^{\frac{-2x}{\max R-CM_i}})^{-1}-1 & x\geqslant CM_i\\ 2(1+e^{\frac{-2x}{CM_i-\min R}})^{-1}-1 & \text{others}\end{cases}$$

$$f_{Uu}(x)=\begin{cases}2(1+e^{\frac{-2x}{\max R-UM_{ui}}})^{-1}-1 & f_{Ii}(x)\geqslant 0\\ 2(1+e^{\frac{-2x}{UM_{ui}-\min R}})^{-1}-1 & \text{others}\end{cases}$$

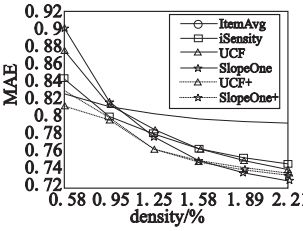
4.3　评估标准

本文采用了传统推荐系统最常用的准确性评估标准平均绝对误差 MAE(mean absolute error)作为评价依据,其他的评价方法可参见文献[5]。

$$\mathrm{MAE}=\frac{1}{n}\sum_{u,i}|\hat{r}_{ui}-r_{ui}|$$

4.4　实验结果及分析

图 2 为 MovieLens 数据集合的结果,图 3 为 Book-Crossing 数据集合的结果。从图 2、3 中可以看出,在 MovieLens 和 Book-Crossing 两个数据集合上,iSensity 比 ItemAvg、UCF、SlopeOne 准确性高;UCF + 、SlopeOne + 比 UCF、SlopeOne 准确性更好,而且性能稳定。



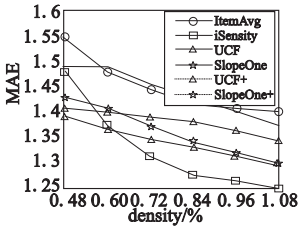


图2　MovieLens数据集合的结果　图3　Book-Crossing数据集合的结果

在 MovieLens 数据集合中,UCF + 、SlopeOne + 的效果比 iSensity 的好,而在 Book-Crossing 数据集合中的效果却不比 iSenstiy 好。其原因可能有两个:a) Book-Crossing 的数据集合中没有项目的分类信息;b) Book-Crossing 的数据集合比 MovieLens 的数据稀疏得多。这说明 iSenstiy 更加适用于评分稀疏的数据集合。

上述实验表明,Sigmoid 函数能够有效对项目的受欢迎程度和用户对项目的喜爱程度进行建模,一定程度上解决了推荐系统中的稀疏性问题。MovieLens 数据集中有项目分类信息,而 Book-Crossing 数据集中却没有,但同样取得了较好的预测效果,这说明了改进算法的普适性。实验中还发现,评分矩阵非常稀疏时 SlopeOne 算法性能比 UCF、ItemAvg 性能差,但其性能随着矩阵密度增长而迅速增加,评分矩阵较稠密时性能超过 UCF,这与文献[12]中的结果一致。

5　结束语

本文针对协同过滤推荐算法存在的稀疏性问题,总结分析了已有的解决方法,并提出了一种利用 Sigmoid 函数对用户和项目进行建模的 iSensity 算法。在两组真实世界测试数据集合上的实验表明,iSensity 在评分矩阵稀疏时是比较有效的,比经典的 ItemAvg、SlopeOne、UCF 算法精度高。利用 Sigmoid 函数对 UCF 和 SlopeOne 的改进算法能够综合全局信息和局部差值结果,在性能方面得到了明显提高,较好地解决了评分矩阵的稀疏性问题。

Sigmoid 函数能够很好地刻画人类的情感分布。如何利用 Sigmoid 函数提取用户潜在的兴趣、建立用户之间的相似度关系,还需要进一步的研究和探索。

参考文献:

[1] ADOMAVICIUS G, TUZHILIN A. Toward 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.

[2] LU Lin-yuan, MEDO M, YEUNG C H, *et al*. Recommender systems [J]. Physics Reports,2012,1(3):159-172.

[3] RICCI F, ROKACH L, SHAPIRA P, *et al*. Recommender systems handbook: a complete guide for scientists and practitioners[M]. [S. l.]:Springer, 2011.

[4] SCHAFER J B, FRANKOWSKI D, HERLOCKER J,*et al*. Collaborative filtering recommender systems[C]//Lecture Notes in Computer Science, vol 4321. Berlin: Springer-Verlag, 2007:291-324.

[5] CACHEDA F, CARNEIRO V, FERNÁNDEZ D, *et al*. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems [J]. ACM Trans on the Web,2011,5(1):1-33.

[6] LEMIRE D, MACLACLAN A. Slope one predictors for online rating-based collaborative filtering[J]. Society for Industrial Mathematics,2005,5:471-480.

[7] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. IEEE Computer,2009,42(8):30-37.

[8] HOFMANN T. Latent semantic models for collaborative filtering[J]. ACM Trans on Information Systems,2004,22(1):89-115.

[9] BLEI D M, NG A Y, JORDAN M I. Latent Dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3 (3/1): 993-1022.

[10] SALAKHUTDINOV R, MNIH A, HINTON G. Restricted Boltzmann machines for collaborative filtering[C]// Proc of the 24th International Conference on Machine Learning. 2007:791-798.

[11] HUANG Zan, ZENG D, CHEN H. A comparison of collaborative filtering recommendation algorithms for e-commerce[J]. IEEE Intelligent Systems,2007,22(5):68-78.

[12] LEE J, SUN Ming-xuan, LEBANON G. A comparative study of collaborative filtering algorithms[C]//Proc of CORR. 2012.

[13] XUE Gui-rong, LIN Chen-xi, YANG Qiang, *et al*. Scalable collaborative filtering using cluster-based smoothing[C]//Proc of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press,2005:114-121.

[14] HUANG Zan, CHEN H, ZENG D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering [J]. ACM Trans on Information Systems, 2004, 22 (1): 116-142.

[15] SUN Duo, ZHOU Tao, LIU Jian-guo, *et al*. Information filtering based on transferring similarity[J]. Physical Review E, 2009, 80 (1):17101.

[16] YILDIRIM H, KRISHNAMOORTHY M S. A random walk method for alleviating the sparsity problem in collaborative filtering[C]// Proc of the 2nd ACM International Conference on Recommender Systems. 2008:131-138.

[17] ESSLIMANI I, BRUN A, BOYER A. Densifying a behavioral recommender system by social networks link prediction methods[J]. Social Network Analysis and Mining,2011,1(3): 159-172.

[18] LIU Qi, CHEN En-hong, XIONG Hui, *et al*. Enhancing collaborative filtering by user interest expansion via personalized ranking[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics,2012,42(1):218-233.

[19] YANG Xi-wang, GUO Yang, LIU Yong. Bayesian-inference based recommendation in online social networks[C]//Proc of IEEE INFOCOM. 2011:551-555.