一种新的社会化相似度计算方法*

李 改[1,2,3]，李章凤[2,3]，李 磊[2,3]
(1. 顺德职业技术学院 电子与信息工程系，广东 顺德 528333；2. 中山大学 信息科学与技术学院，广州 510006；3. 中山大学 软件研究所，广州 510275)

摘 要：基于社交网络的推荐算法引入社交网络信息到协同过滤算法中来，使得用户朋友的偏好能够影响用户本身的偏好 。这些算法需要用到用户之间的相似度信息。目前有两个流行的基于共同评分项目集的相似度计算函数(VSS、PCC)。在很多情况下，由于用户间没有共同评分项目集，故无法计算他们之间的相似度。为了解决这个问题，提出了一种基于矩阵分解的新的社会化相似度计算方法。在真实的包含社交网络的数据集上进行实验验证，该方法的性能优于几个经典的基于社交网络的协同过滤算法，而且能够解决新用户的冷启动问题。

关键词：推荐系统；协同过滤；社交网络；冷启动
中图分类号：TP311 **文献标志码**：A **文章编号**：1001-3695(2013)09-2616-03
doi:10.3969/j.issn.1001-3695.2013.09.014

New social similarities for collaborative filtering

LI Gai[1,2,3], LI Zhang-feng[2,3], LI Lei[2,3]
(1. *School of Electronics & Information Engineering*, *Shunde Polytechnic*, *Shunde Guangdong* 528333, *China*; 2. *School of Information Science & Technology*, *Sun Yat-sen University*, *Guangzhou* 510006, *China*; 3. *Software Institute*, *Sun Yat-Sen University*, *Guangzhou* 510275, *China*)

Abstract: Some recommendation methods based on social network have been raised. These approaches incorporate social network into the CF algorithms and users' preferences can be influenced by the favors of their friends. These social approaches require the knowledge of similarities among friends. There are two popular similarity functions: vector space similarity (VSS) and Pearson correlation coefficient (PCC). However, both friends similarity functions are based on the item-sets they rated in common. In most cases, these functions are impractical, i. e. if two friends do not share the same items in common, the similarity between them will be zeros. To solve this problem, this paper proposed a new social similarity (NSS) function based on the matrix factorization technique. It conducted experiment on a large dataset: Epinions, which was a widely-used dataset with social information. The experiment results illustrate that this approach outperforms the baseline models and achieves a better performance than social-based method, and can solve the problem of new users.

Key words: recommended systems; collaborative filtering; social network; cold start

协同过滤是推荐系统中运用最广泛的推荐技术[1~4]。协同过滤算法的核心是分析用户兴趣，在用户群中找到与指定用户相似(兴趣)的用户，综合这些相似用户对某一信息的评价，形成系统对该用户对此信息的喜好程度预测。然而，传统的协同过滤算法没有考虑用户之间的社会化关系。例如：当人们想看电影的时候，更倾向于要朋友来推荐电影而不是询问兴趣相似的陌生人。传统的协同过滤算法还面临新用户的冷启动问题。新用户是指那些没有评过分的用户或评分比较少的用户，因为系统中该用户的信息量太少，所以无法对其提供有效的推荐服务。

当前随着互联网上社交网络的出现，使得如何有效利用社交网络信息来提高推荐性能成了新的研究课题。现在已经有一些包含了社交网络信息的推荐算法被提出，这些算法运用用户朋友的评分信息来提高该用户的被推荐效果[5~8]。社交网络信息的另一个用途在于它可以用于解决新用户的冷启动问题。例如，尽管一个卓越网的新用户可能以前没有购买任何商品，但是如果知道新用户朋友的信息，也可以通过学习新用户朋友的偏好来为该新用户进行有效推荐。

Ma 等人[5]提出运用向量空间相似度(VSS)和皮尔生相关系数(PCC)作为朋友间的相似度度量。但运用这两个函数计算相似度也存在两个问题，即朋友丢失和新用户的冷启动问题。为此，本文提出了一种新的社会化相似度计算函数。

1 基本定义和相关工作

1.1 基本定义

本文中给定的矩阵 R 表示具有 m 个用户、n 个对象的评分矩阵，矩阵 U、V 分别表示用户和推荐对象的特征矩阵。

1.2 相关工作

协同过滤算法主要包括两类，一类是基于内容的算法，一类是基于模型的算法。尽管协同过滤算法在实际的推荐系统中取得了巨大成功，但也存在一些问题限制了它的应用。最大的问题就是数据稀疏性问题。为了解决这个问题，一些研究提出运用社交网络信息来提高推荐精度。这些算法通过寻找相

收稿日期：2012-12-17；**修回日期**：2013-02-25 **基金项目**：国家自然科学基金资助项目(61003140,61033010)；中山大学高性能与网格计算平台资助项目

作者简介：李改(1981-)，男，湖北荆州人，博士研究生，主要研究方向为数据挖掘、推荐系统(ligai999@126.com)；李章凤(1989-)，男，硕士研究生，主要研究方向为数据挖掘、推荐系统；李磊(1951-)，男，教授，博导，博士，主要研究方向为数据库、数据挖掘和人工智能.

似的用户来产生推荐。在文献[5]中,Ma 等人提出运用 VSS 和 PCC 来计算用户之间的相似度,从而进行相应的社会化推荐。由于这两个相似度计算方法均基于用户之间的共同评分项目集,而在现实世界中可能存在两个用户各自都有很多评分项,但这两个用户之间没有共同评分项目集,因此不能运用这两个函数来计算相应的用户间的相似度,从而导致这两个用户之间的社交网络信息丢失。为了解决该问题,本文提出了一种新的社会化相似度计算函数,使其能计算没有共同评分的用户间的相似度,同时解决了新用户的冷启动问题。

2　基于新相似度计算方法的社会化推荐算法

2.1　传统的矩阵分解算法

基于矩阵分解的协同过滤算法是最流行、运用最广泛的协同过滤推荐算法。在传统的基于矩阵分解的协同过滤推荐算法中,希望找到一个低秩矩阵 X 来逼近矩阵 R。其中 $X = UV^{\mathrm{T}}$, $U \in C^{m\times d}$, $V \in C^{n\times d}$, d 表示特征个数,一般 $d << r$, r 表示矩阵 R 的秩, $r \leqslant \min(m,n)$。

为了找到一个低秩矩阵 X 来最大程度地逼近矩阵 R。最小化下面的 Frobenius 损失函数:

$$L(X) = \sum_{ij} I_{ij}(R_{ij} - X_{ij})^2 \tag{1}$$

其中:$(R_{ij} - X_{ij})^2$ 是低秩逼近中常见的平方误差项;$I = (I_{ij})_{m\times n}$,如果 $R_{ij} \neq 0$,则 $I_{ij} = 1$,否则 $I_{ij} = 0$。下面考虑如何有效并且快速地求解最优化问题 $\arg\min_X L(X)$。

式(1)可以改写为

$$L(U,V) = \sum_{ij} I_{ij}(R_{ij} - U_{i.}V_{j.}^{\mathrm{T}})^2 \tag{2}$$

为了防止过拟合,式(2)加上正则化项,则式(2)可改写为

$$L(U,V) = \sum_{ij} I_{ij}(R_{ij} - U_{i.}V_{j.}^{\mathrm{T}})^2 + \lambda(\|U_{i.}\|_F^2 + \|V_{j.}\|_F^2) \tag{3}$$

2.2　基于社交网络的矩阵分解算法

互联网进入了社交网络时代,人们更倾向于询问朋友来获得推荐服务。因为用户的喜好在很大程度上可能与他的朋友相似。所以在传统的基于矩阵分解的协同过滤算法上,给式(3)加上社会化正则项:

$$L(U,V) = \sum_{ij} I_{ij}(R_{ij} - U_{i.}V_{j.}^{\mathrm{T}})^2 + \lambda(\|U_{i.}\|_F^2 + \|V_{j.}\|_F^2) + \frac{\beta}{2}\sum_{f\in F(i)} \mathrm{sim}(i,f)\|U_{i.} - U_{f.}\|_F^2 \tag{4}$$

这里添加的社会化正则项是为了使得用户的特征向量尽可能地与他的朋友特征向量相似。参数 β 用于约束社交网络信息的影响;$F(i)$ 是用户 u_i 的朋友集;$\|U_{i.} - U_{f.}\|_F^2$ 用于表示某个用户和他的朋友之间的偏好差异,用户 u_i 和他的朋友 u_f 之间的偏好差异越大,则 $\|U_{i.} - U_{f.}\|_F^2$ 越大;$\mathrm{sim}(i,f)$ 表示用户 u_i 和他的朋友 u_f 之间的相似度;$\mathrm{sim}(i,f)\|U_{i.} - U_{f.}\|_F^2$ 表示用户 u_i 和他的朋友 u_f 通过相似度加权后的偏好差异。

为了最小化损失函数,在式(4)上,通过对 U_i 和 V_j 实施梯度下降法,得到

$$\frac{\partial L^2}{\partial U_i} = \sum_{j=1}^{n} I_{ij}(R_{ij} - U_{i.}V_{j.}^{\mathrm{T}})V_{j.} + \lambda U_{i.} + \beta\sum_{f\in F(i)} \mathrm{sim}(i,f)(U_{i.} - U_{f.}) \tag{5}$$

$$\frac{\partial L^2}{\partial V_j} = \sum_{i=1}^{m} I_{ij}(R_{ij} - U_{i.}V_{j.}^{\mathrm{T}})U_{i.} + \lambda V_{j.} \tag{6}$$

2.3　新的相似度计算方法

在式(4)~(6)中,$\mathrm{sim}(i,f)$ 用于计算用户 u_i 和他的朋友 u_f 之间的相似度。由于 VSS 和 PCC 均基于用户之间的共同评分项目集来计算相似度,而在现实世界中可能存在两个用户各自都有很多评分项,但这两个用户之间没有共同评分项目集,因此不能运用这两个函数来计算相应的用户间的相似度,从而导致这两个用户之间的社交网络信息丢失。为了解决该问题,本文提出了一种新的社会化相似度计算函数 NSS 使其能计算没有共同评分的用户间的相似度:

$$\mathrm{sim}(i,f) = \cos(U_{i.}, U_{f.}) = \frac{\sum_{k=1}^{d} U_{ik}U_{fk}}{\sqrt{\sum_{k=1}^{d} U_{ik}^2}\sqrt{\sum_{k=1}^{d} U_{fk}^2}} \tag{7}$$

其中:$U_{i.}$ 和 $U_{f.}$ 表示两个用户 u_i 和 u_f 的潜在特征向量。因此,如果用户 u_i 和 u_f 偏好相同,则特征向量 $U_{i.}$ 和 $U_{f.}$ 也相近,从而计算出来的两个用户 u_i 和 u_f 的相似度 $\mathrm{sim}(i,f)$ 接近于 1。如果运用 VSS 和 PCC 来计算相似度,将导致没有共同评分项目集的用户之间的社会化信息丢失。因为本文提出的 NSS 相似度计算方法不需要用户之间的共同评分项目集,所以不会导致用户之间的社会化信息丢失。

3　实验结果及分析

本章首先介绍本文实验所采用的数据集及评价标准,然后给出本文所提出的基于新的相似度计算方法的社会化协同过滤算法与几个传统的矩阵分解算法和社会化协同过滤算法的实验结果比较。

3.1　实验数据集

在本实验中使用广泛运用的、公用的包含社交网络数据的数据集 Epinions[9]。在这个数据集中,用户对其感兴趣的产品和服务进行评分,评分范围是 1~5 分的整数值。Epinions 数据集中有用户的社交网络信息,这些关系以用户所信任的朋友的关系存在。Epinions 数据集中包含 49 290 个用户,139 738 个不同的评分项。总的评分数据点是 664 824 个。在社交网络中,总共有 511 799 个信任关系。

3.2　实验的评价标准

本文实验采用 RMSE 作为评价标准[5],RMSE 通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性。RMSE 为推荐质量提供了直观的度量方法,是最常用的一种推荐质量度量方法。推荐算法整体的 RMSE 越小,意味着推荐的质量越高。假设算法对 N 个项目预测的评分向量表示为 $\{p_1, p_2, \cdots, p_N\}$,对应的实际用户评分集合为 $\{r_1, r_2, \cdots, r_N\}$,则算法的 RMSE 表示为

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{N}(p_i - r_i)^2/N} \tag{8}$$

3.3　实验结果

3.3.1　新算法与其他协同过滤算法的实验结果比较

将本文所提出的基于新的相似度计算方法的社会化协同过滤算法与几个传统的矩阵分解算法和社会化协同过滤算法的实验结果进行比较。

PMF 表示传统的概率矩阵分解算法,由 Salakhutdinov 等人[10]提出,它仅仅使用评分矩阵来进行推荐。SR_1 表示传统的基于社交网络的协同过滤算法,其朋友之间的相似度设为固定值 1,也就是设置 $\mathrm{sim}(i,f) = 1$。$\mathrm{SR}_{\mathrm{PCC}}$ 表示传统的基于社交网络的协同过滤算法,其朋友之间的相似度运用 PCC 来计

算[5]。NSS 表示用本文所提出的相似度计算方法来计算朋友间相似度的基于社交网络的协同过滤推荐算法。结果如表 1 所示。

表 1 NSS 算法和几个经典的 CF 算法的性能比较(RMSE)

特征数	算法			
	PMF	SR_1	SR_{PCC}	NSS
$d=5$	1.1694	1.1487	1.1351	1.1031
$d=10$	1.1765	1.1579	1.1436	1.1156

实验结果显示,与传统的概率矩阵分解算法 PMF 相比,当特征数为 5 时,算法 NSS 的性能提高了 5.67%,当特征数为 10 时,算法 NSS 的性能提高了 5.18%。与采用 PCC 作为朋友之间的相似度度量的 SR_{PCC}算法相比,当特征数为 5 时,算法 NSS 的性能提高了 2.82%,当特征数为 10 时算法 NSS 的性能提高了 2.44%。从表 1 可以看出,当特征数 $d=5$ 时,算法性能已经足够好。

3.3.2 新用户情况下新算法与其他协同过滤算法的实验结果比较

本文算法能够有效解决新用户的冷启动问题。在本实验中,定义评分少于 10 个评分点的用户为新用户。传统的概率矩阵分解算法 PMF 只使用了评分数据,故其不能解决新用户的冷启动问题。从表 2 中看到,算法 SR_1 的性能优于算法 PMF 的性能,因为其使用了社交网络信息。算法 SR_{PCC}的性能优于算法 SR_1,因为算法 SR_{PCC}使用 PCC 计算朋友之间的相似度比把相似度固定为 1 更可靠。算法 NSS 的性能优于算法 SR_{PCC},因为算法 NSS 使用用户的特征向量来计算朋友之间的相似度,不依赖于朋友之间的共同评分项目集,所以不会导致算法 SR_{PCC}所面临的新用户和他的朋友之间的社会化信息丢失的问题,能够更好地解决新用户的冷启动问题。

表 2 新用户下 NSS 算法和几个经典的 CF 算法的性能比较(RMSE)

特征数	算法			
	PMF	SR_1	SR_{PCC}	NSS
$d=5$	1.1970	1.1833	1.1629	1.1489
$d=10$	1.1947	1.1811	1.1606	1.1454

新用户的社交网络信息在新用户和其活跃的朋友之间建立起了某种联系,新用户的偏好在很大程度上可能与他的朋友的偏好相似。因此通过使用本文所提出的新的相似度计算方法引入社交网络信息能够解决新用户的冷启动问题,而不会导致社会化信息丢失。

从表 2 中不难看出,当特征数为 5 时,与算法 PMF 相比,算法 NSS 的性能提高了 4.02%。因为新用户没有很多的评分信息可以使用,朋友关系也可能比较少。所以在这里算法 NSS 的性能提高幅度没有使用完整的 Epinions 数据集时多,使用完整的 Epinions 数据集时算法 NSS 的性能优于算法 PMF 5.67%。

3.3.3 参数 λ 和 β 对算法性能的影响分析

参数 λ 是正则化参数,可以通过交叉确认的方法学习得到,在本文中设置该参数为 0.01。另一个参数 β 用于控制社交网络信息对推荐系统性能的影响。在极端情况下,如果设置参数 $\beta=0$,则在推荐过程中社交网络信息没有产生任何作用,推荐算法退化为传统的基于矩阵分解的推荐算法。

图 1 显示了当特征数为 5 时,参数 β 对算法 NSS 性能的影响,其中横轴表示参数 β 值的变化,纵轴表示算法 NSS 的 RMSE 值。从图 1 不难看出,随着参数 β 的增大,算法 NSS 的 RMSE 值首先急剧下降,随后又逐渐升高,参数 β 的最优值是 0.01。这个结果正好和现实世界中朋友之间相互推荐的实际效果相符合。如果既考虑自身的喜好又考虑朋友的推荐,则就更能找到符合自己偏好的推荐对象。如果仅仅只是考虑自身喜好而忽略朋友的推荐,或者是仅仅考虑朋友的推荐而忽略自身的喜好,则都会走向极端,而不能得到最好的推荐效果。

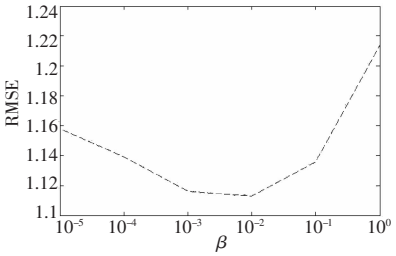

图 1 参数 β 对算法 NSS 性能的影响

3.3.4 新算法与其他协同过滤算法的运行时间比较

本文所有算法均在装有 Linux 系统环境的计算机下运行,计算机配置环境是 4 核 CPU,主频是 3.2 GHz,内存是 7.6 GB。各个算法的程序代码用 Java 开发工具 Eclipse 编写。每个算法均循环迭代 100 轮。各个算法均反复运行 10 次取平均时间作为该算法的最终运行时间。

从表 3 可以看出,算法 PMF 在各个特征数下的运行时间均小于其他算法,这是因为 PMF 没有考虑社交网络的信息,故运算更快。在考虑了社交网络的其他三个算法中,NSS 在各个特征数下的运行时间明显优于 SR_{PCC}和 SR_1。这主要是由于本文所提出的不依赖于共同评分的相似度计算方法,在计算式(5)(6)的同时,可以实时更新用户朋友之间相似度,故节省了朋友之间的相似度的计算时间。而 SR_{PCC}在计算式(5)(6)之前就需要提前计算好用户朋友间的相似度,且运用 PCC 来计算相似度比运用式(7)来计算相似度复杂度更高。SR_1 在各个特征数下的运行时间均小于 NSS,主要是因为在 SR_1 中用户朋友间的相似度统一设置为 1,无须提前计算,故运行时间大幅缩减。从表 3 还可看出,各个算法在特征数为 10 时的运行时间均大于特征数为 5 时的运行时间,这主要是因为随着特征数的增加,计算式(5)(6)所需时间更多。

表 3 NSS 算法和几个经典的 CF 算法的运行时间比较 /s

特征数	算法			
	PMF	SR_1	SR_{PCC}	NSS
$d=5$	11.675	20.854	2007.392	30.558
$d=10$	14.008	24.303	3278.044	33.463

4 结束语

本文提出了一种基于矩阵分解的新的社会化相似度计算方法。在真实的包含社交网络数据集上的实验验证了该方法能够更好地计算用户朋友之间的相似度,在 RMSE 评价指标下性能均优于几个经典的基于社交网络的协同过滤算法,运行时间相较经典的 SR_{PCC}算法也有显著下降。而且由于该相似度的计算不基于朋友间的共同评分项目集,因此能够用于解决新用户的冷启动问题。在以后的研究中,笔者将运用本文所提出的新的社会化相似度计算方法去解决用户朋友间的信任传递问题,同时将更关注所提算法的排序性能。

参考文献:

[1] PAN Rong, ZHOU Yun-hong, CAO Bin, *et al*. One-class collaborative filtering[C]//Proc of the 8th IEEE ICDM. Washington DC: IEEE Computer Society, 2008: 502-511.

(下转第 2644 页)

表 3 在目标精度下的平均迭代次数与成功率比较

函数	维数	目标精度	测试项目	FOA	FOAAM	SPSO	LDWPSO	APSO
f_1	30	10^{-5}	平均迭代次数	1402(1)	1416(1)	528	2183	484
			成功率	0.75 (1)	1 (1)	1	1	1
f_2	30	10^{-6}	平均迭代次数	134(1)	381(1)	413	2133	454
			成功率	0.7(1)	1(1)	0.74	1	1
f_3	30	28.8	平均迭代次数	25(1)	26(1)	911	2477	994
			成功率	0.85(1)	1(1)	1	1	1
f_4	30	0	平均迭代次数	∞ (1)	82(1)	196	1710	420
			成功率	0 (1)	1 (1)	0.58	1	1
f_5	30	10^{-1}	平均迭代次数	∞ (∞)	∞ (∞)	∞	2197	412
			成功率	0(0)	0(0)	0	0.72	1
f_6	2	−1	平均迭代次数	∞	71	−	−	−
			成功率	0	1	−	−	−

表 4 高维、多峰函数上的算法性能比较

Function	Test Item	FOA	FOAAM	GA	BFO	PSO	BFA-PSO
f_2	优化均值 (标准差)	2.4827e−6 (3.6032e−6)	7.7156e−7 (5.0379e−7)	1.505 (1.15)	1.028 (0.8e−2)	4.2353e+1 (1.18556e+1)	4.58e−3 (3.4372e−3)
f_4	优化均值 (标准差)	2.2352e−4 (4.3320e−5)	0 (0)	66.791 (11.57)	3.73 (3.6e−1)	2.4382e+2 (35.1271)	4.9337e+1 (8.8406)
f_5	优化均值 (标准差)	1.6852 (7.8881e−5)	1.6844 (1.7200e−15)	4.262 (8.77e−1)	3.97e−1 (3.4e−2)	1.35e+1 (9.68e−1)	1.128e−1 0.0111

由上可知,本文提出的 FOAAM 总体来说比 FOA、文献[7,11,12]中算法具有更好的全局搜索能力,更快的收敛速度和更高的收敛精度。

4 结束语

本文针对果蝇优化算法寻优精度不高和易陷入局部最优的缺点,提出了一种自适应变异的果蝇优化算法。六个经典测试函数的对比实验结果表明,新算法具有更好的全局搜索能力,在收敛速度、收敛精度及收敛可靠性上均比基本果蝇优化算法有较大的提高。

参考文献:

[1] PAN Wen-tsao. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. Knowledge-based Systems, 2012, 26: 69-74.

[2] 潘文超. 果蝇最佳化演算法[M]. 台北: 沧海书局, 2011: 10-12.

[3] 潘文超. 应用果蝇优化算法优化广义回归神经网络进行企业经营绩效评估[J]. 太原理工大学学报: 社会科学版, 2011, 29(4): 1-5.

[4] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861-868.

[5] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420.

[6] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001: 148-149.

[7] 林川, 冯全源. 一种新的自适应粒子群优化算法[J]. 计算机工程, 2008, 34(7): 181-183.

[8] LI M S, JI T Y, TANG W J, *et al*. Bacterial foraging algorithm with varying population[J]. Biosystems, 2010, 100(3): 185-197.

[9] 靳其兵, 赵振兴, 苏晓静, 等. 基于粒子健康度的快速收敛粒子群优化算法[J]. 化工学报, 2011, 62(8): 2328-2333.

[10] 阳春华, 谷丽姗, 桂卫华. 自适应变异的粒子群优化算法[J]. 计算机工程, 2008, 34(16): 188-190.

[11] 胡洁. 细菌觅食优化算法的改进及应用研究[D]. 武汉: 武汉理工大学, 2012.

[12] 杨萍, 孙延明, 刘小龙, 等. 基于细菌觅食趋化算子的 PSO 算法[J]. 计算机应用研究, 2011, 28(10): 3640-3642.

(上接第 2618 页)

[2] PAN Rong, SCHOLZ M. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering[C]//Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 667-676.

[3] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extentions[J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[4] ZHANG Yu, CAO Bin, YEUNG D Y. Multi-domain collaborative filtering[C]//Proc of the 26th Conference on Uncertainty in Artificial Intelligence. 2010: 725-732.

[5] MA Hao, ZHOU Deng-yong, LIU Chao, *et al*. Recommender systems with social regularization[C]//Proc of the 4th International Conference on WSDM. New York: ACM Press, 2011: 287-296.

[6] MA Hao, YANG Hai-xuan, LYU M R, *et al*. SoRec: social recommendation using probabilistic matrix factorization[C]//Proc of the 17th ACM Conference on Information and Knowledge Management. New York: ACM Press, 2008: 931-940.

[7] MA Hao, KING I, LYU M R. Learning to recommend with social trust ensemble[C]//Proc of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2009: 203-210.

[8] JAMALI M, ESTER M. Matrix factorization technique with trust propagation for recommendation in social networks[C]//Proc of the 4th ACM Conference on Recommender Systems. New York: ACM Press, 2010: 135-142.

[9] MASSA P, AVESANI P. Trust metrics in recommender systems[M]// Computing with Social Trust. London: Springer, 2009: 259-285.

[10] SALAKHUTDINOV R, MNIH A. Probabilistic matrix factorization [C]//Proc of NIPS. 2008: 1257-1264.