

基于个性化情境和项目类别的资源推荐研究

杨 畅 李 华

(重庆大学计算机学院 重庆 400044)

摘 要 传统的协同过滤推荐技术没有考虑影响用户评分的用户情境信息,但最近研究发现用户个性化情境信息直接影响着用户评分,因此在传统的协同过滤技术基础上引入用户个性化情境后推荐效果有所提高。此外可以将用户个性化情境和项目类别相结合起来。先对项目进行分类,然后再确定用户在每个项目类别下的个性化情境,同一项目类别下所有项目的用户个性化情境是相同的。在为目标项目预测评分时,先确定目标项目所在的类别,进而确定计算目标项目预测评分所用到的用户个性化情境。实验结果表明,改进后的算法较 Slope one 有较大提高。

关键词 协同过滤,个性化情境,项目类别,推荐

Personalize Context and Item Class Based Resource Recommendation

YANG Chang LI Hua

(College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract The conditional collaborative filtering technology does not consider the user's context information which affect user's rating. But recent research show that user's personalized context directly affect rating, so the result of recommendation can be improved if personalize context is incorporated into conditional collaborative filtering technology. Besides, the personalized context and item class are can be combined, Firstly classifying the items, and then making sure user's personalize context under every item class. When predicting the rating of target item, Firstly make sure which item class the target item is belong to, and then identify the user's personalized context used to compute the rating of the target item. The experimental results show that the recommendation accuracy of proposed approach is better than Slope One.

Keywords Collaborative filtering, Personalized context, Item class, Recommendation

随着互联网的发展,网上的资源越来越多,个性化的资源推荐开始越来越受人们的关注。涌现出了多种推荐方法,其中协同过滤(collaborative filtering)是最具代表性,在应用中使用最广泛的推荐技术。协同过滤可分为两种:基于用户的协同过滤^[1](user-based collaborative filtering)和基于项目的协同过滤^[2](items-based collaborative filtering)。基于用户的协同过滤的基本思想是:向目标用户推荐项目前先要确定目标用户的邻居用户,通过邻居用户对项目的评分来预测目标用户对该项目的评分。目标用户对邻居用户评价高的项目会感兴趣。这种推荐技术对用户的评分数据依赖很大,而往往用户对项目的评分率较低^[3],所以这种推荐技术存在数据稀疏性、冷启动的问题,而且随着用户和项目数量的增多,推荐效率明显下降。基于项目协同过滤解决了因计算邻居用户的实时性随用户增多急剧下降的问题,其基本思想是:预测用户对目标项目的评分前先确定目标项目的邻居项目,通过邻居项目的评分来预测目标项目的评分。项目之间的相似性比较稳定,从而在一定时间内更新一次,从而明显改善对实时性的要求,但在预测的准确性方面并没有很大的提高。

对此,研究人员在此基础上提出了一些改进的方法。如传统 Items-Based 协同过滤推荐算法改进、基于项目评分预测的协同过滤推荐算法、基于领域最近邻的协同过滤算法、基于个性化情境和项目的协同推荐研究等。其中文献[4]提出将

个性化情境和基于项目的协同过滤结合起来,将项目的相似性比较放在某一情境下,不同的情境下项目的相似性不同,这样做的好处是:提高了项目相似性计算的准确性,减少了计算量,提高了项目评分的准确度。文献[7]提出根据用户对项目评分的相似性对项目进行聚类,即提出了项目分类的概念。本文在此基础上提出了将个性化情境和项目分类相结合的方法,首先将项目进行分类产生不同的项目类,再在项目类中引入个性化情境,进一步缩小比较项目相似性的范围,提高项目预测的准确性。前者是在整体项目下引入个性化情境,并没有考虑项目本身的属性。实验也证明本方法确实可行。

1 相关工作

大多数协同过滤算法都是基于用户-项目评分矩阵 R 进行的。 R 是一个 $m \times n$ 的矩阵,其中 m 行代表 m 个用户, n 列代表 n 个项目, $R_{i,j}$ 表示用户 i 对项目 j 的评分值(见表1)。

表1 用户-项目评分矩阵 $R(m,n)$

	item ₁	...	item _j	...	item _n
user ₁	$R_{1,1}$	...	$R_{1,j}$	...	$R_{1,n}$
⋮	⋮	⋮	⋮	⋮	⋮
user _i	$R_{i,1}$	...	$R_{i,j}$	...	$R_{i,n}$
⋮	⋮	⋮	⋮	⋮	⋮
user _m	$R_{m,1}$	...	$R_{m,j}$	...	$R_{m,n}$

1.1 基于用户的协同过滤

基于用户的协同过滤推荐是根据用户-项目评分矩阵来寻找与目标用户最相似的前 p 个用户作为邻居用户,然后根据邻居用户的评分来计算目标用户对未评分项目的预测评分。可以看出,计算用户的相似性是关键。当前度量用户相似性的方法有:余弦相似性、相关相似性、修正的余弦相似性和条件概率等^[2]。

(1)余弦相似性

$$\text{sim}(u, v) = \cos(u, v) = \frac{\sum_{i \in I} R_{u,i} \times R_{v,i}}{\sqrt{\sum_{i \in I} R_{u,i}^2} \times \sqrt{\sum_{i \in I} R_{v,i}^2}} \quad (1)$$

式中, I 表示项目总集, $i \in I$ 表示 I 中的任一项目 i , $R_{u,i}$ 和 $R_{v,i}$ 分别表示用户 u 和用户 v 对项目 i 的评分。

(2)相关相似性

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u) \times (R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)^2} \times \sqrt{\sum_{i \in I_{u,v}} (R_{v,i} - \bar{R}_v)^2}} \quad (2)$$

式中, I 表示项目总集, $i \in I$ 表示 I 中的任一项目 i , $R_{u,i}$ 和 $R_{v,i}$ 分别表示用户 u 和用户 v 对项目 i 的评分, $\bar{R}_u$ 和 $\bar{R}_v$ 分别表示用户 u 和用户 v 对项目的评分的平均值。 $I_{u,v}$ 表示用户 u 和用户 v 共同评分的项目。

(3)修正的余弦相似性

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u) \times (R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_u} (R_{u,i} - \bar{R}_u)^2} \times \sqrt{\sum_{i \in I_v} (R_{v,i} - \bar{R}_v)^2}} \quad (3)$$

式中, I 表示项目总集, $i \in I$ 表示 I 中的任一项目 i , $R_{u,i}$ 和 $R_{v,i}$ 分别表示用户 u 和用户 v 对项目 i 的评分, $\bar{R}_u$ 和 $\bar{R}_v$ 分别表示用户 u 和用户 v 对项目的评分的平均值。 $I_{u,v}$ 表示用户 u 和用户 v 共同评分的项目, I_u 和 I_v 分别表示用户 u 和用户 v 评分的项目。

1.2 基于项目的协同过滤

基于项目的协同过滤和基于用户的协同过滤在思想上类似,只是出发角度不同,前者是从项目的角度出发,后者是从用户的角度出发。基于项目的协同过滤推荐是根据用户-项目评分矩阵寻找与目标项目最相似的前 k 个项目,然后根据用户对这最相似的 k 个项目的评分来预测用户对目标项目的评分。因为项目之前的相似性比较稳定,所以求项目间的相似性和寻找项目的前 k 个最相似的项目都可以离线计算,相对基于用户的协同过滤来说,可扩展性大大提高。当前度量项目相似性的方法类似于计算用户相似性,有以下 3 种:余弦相似性、相关相似性、修正的余弦相似性。

(1)余弦相似性

$$\text{sim}(i, j) = \cos(i, j) = \frac{\sum_{u \in U} R_{u,i} \times R_{u,j}}{\sqrt{\sum_{u \in U} R_{u,i}^2} \times \sqrt{\sum_{u \in U} R_{u,j}^2}} \quad (4)$$

式中, U 表示用户总集, $u \in U$ 表示 U 中的任一用户 u , $R_{u,i}$ 和 $R_{u,j}$ 分别表示用户 u 对项目 i 和 j 的评分。

(2)相关相似性

$$\text{sim} = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - \bar{R}_i) \times (R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{i,j}} (R_{u,i} - \bar{R}_i)^2} \times \sqrt{\sum_{u \in U_{i,j}} (R_{u,j} - \bar{R}_j)^2}} \quad (5)$$

式中, U 表示用户总集, $u \in U$ 表示 U 中的任一项目 u , $R_{u,i}$ 和 $R_{u,j}$ 分别表示用户 u 对项目 i 和 j 的评分, $\bar{R}_i$ 和 $\bar{R}_j$ 分别表示所

有用户对项目 i 和 j 的评分的平均值。 $U_{i,j}$ 表示项目 i 和 j 共同评分的用户。

(3)修正的余弦相似性

$$\text{sim} = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - \bar{R}_i) \times (R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U_i} (R_{u,i} - \bar{R}_i)^2} \times \sqrt{\sum_{u \in U_j} (R_{u,j} - \bar{R}_j)^2}} \quad (6)$$

式中, U 表示用户总集, $u \in U$ 表示 U 中的任一用户 u , $R_{u,i}$ 和 $R_{u,j}$ 分别表示用户 u 对项目 i 和 j 的评分, $\bar{R}_i$ 和 $\bar{R}_j$ 分别表示所有用户对项目 i 和 j 的评分的平均值。 $U_{i,j}$ 表示项目 i 和 j 共同评分的用户。 U_i 和 U_j 分别表示项目 i 和 j 评分的用户。

2 基于个性化情境和项目分类的资源推荐

情境信息是指人的行为或事件发生影响的上下文信息或场景信息,如时间、地点等^[6]。现实中用户对项目的评分综合了用户自身多种情境信息,因此在预测用户对目标项目的评分时可以引入用户的情境信息,在这一方面文献[4]中提出了个性化情境,就情境信息对用户评分影响做了较深入的研究,其推荐算法的准确性较传统的协同过滤有进一步的提高。该文献中对所有项目的预测评分用到的都是同样的个性化情境,事实上我们对用户、用户情境和项目之间的关系进一步研究发现可以对项目先进行分类,然后以项目类为基础引入个性化情境,对同一类项目的预测评分采用同样的个性化情境。提出这个想法是基于这样一个事实:用户对同一类项目的评分比较接近。

2.1 项目差异性矩阵的生成

第一步 项目分类,根据项目的相关属性将项目分成 n 类。例如电影={动作片、爱情片、儿童片、冒险片、科幻片、战争片、悬疑片、恐怖片}。

第二步 将评分矩阵 R 根据项目类别 $S = \{s_1, s_2, \dots, s_n\}$ 划分成子评分矩阵 $R_1, R_2, \dots, R_n$ 。

第三步 根据用户的不同情境信息 $C = \{c_1, c_2, \dots, c_n\}$ 将第二步中生成的子评分矩阵 R_k 分成 $R_{c_1}^k, R_{c_2}^k, \dots, R_{c_n}^k$ 。

第四步 提取第三步矩阵 $R_{c_k}^k$ 中任意 2 项目 i 和 j 的共同评分用户 $U_{i,j}$ 和评分值 $R_{u,i}$ 和 $R_{u,j}$ 。

第五步 计算项目 i 和 j 的评分差异值 $d_{i,j}$:

$$d_{i,j} = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - R_{u,j})}{|U_{i,j}|} \quad (7)$$

式中, $|U_{i,j}|$ 表示对项目 i 和项目 j 共同评分的用户个数,由 $d_{i,j}$ 构成 R_k^k 下的项目差异性矩阵 $D_{c_k}^k$ 。

2.2 用户在项目类下的个性化情境的生成

用户在项目类下的个性化情境的生成即为确定影响用户对某类项目评分最大的个性化情境。

第一步 将评分数据的 80% 作为训练集 D , 将评分数据的 20% 作为测试集 T 。

第二步 将训练集 D 和测试集 T 生成训练评分矩阵 R_d 和测试评分矩阵 R_t , 并合并 R_d 和 R_t 为 R , 然后根据 2.1 节中的第二步和第三步将 R 按项目分类 s_k 和用户情境信息 $C = \{c_1, c_2, \dots, c_n\}$ 划成评分矩阵 $R_{c_1}^k, R_{c_2}^k, \dots, R_{c_n}^k$ 。

第三步 按照 2.1 节中的第四步和第五步生成相应的项目差异性矩阵 $D_{c_1}^k, D_{c_2}^k, \dots, D_{c_n}^k$ 。

第四步 根据训练集 D 和第二步生成的项目差异性矩

阵利用以下公式计算用户对测试集 T 中项目的预测评分。

$$p_{u,i} = \frac{\sum_{j \in I_u} R_{u,j} - d_{j,i}}{|I_u|} \quad (8)$$

式中, $p_{u,i}$ 表示用户 u 对项目 i 的预测评分, I_u 表示训练评分矩阵 R_d 中用户评分的项目, $R_{u,j}$ 表示用户 u 对项目 j 的评分, $d_{i,j}$ 表示项目 i 和 j 的项目差异值, $|I_u|$ 表示训练评分矩阵 R_d 中用户 u 评分的项目个数。

第五步 根据测试集 T 中用户 u 对项目的真实评分和第四步中计算得到的预测评分, 按照式(9)计算它们的 MAE 值。

$$MAE = \frac{\sum_{i=1}^N p_i - R_i}{N} \quad (9)$$

式中, p_i 表示预测评分, R_i 表示真实评分, N 表示用户 u 预测的项目个数。

第六步: 取 MAE 值最小的用户情境 c_k 为用户在项目类别 s_k 中的个性化情境。

2.3 基于个性化情境和项目分类的预测评分

输入: 用户 u 、待预测的项目 i 、训练集 D 。

输出: 用户 u 对项目 i 的预测评分值 $p_{u,i}$ 。

第一步 确定待预测的项目 i 的所属项目类别 s_k 。

第二步: 根据 2.2 节中的计算(离线计算)得到用户 u 在项目类别 s_k 下的个性化情境 c_k 。

第三步 根据个性化情境 c_k 查找具有用户情境 c_k 的用户集 U 。

第四步 查找用户集 U 中的用户对待预测的项目的评分, 生成单列的评分矩阵 R_i , 根据训练集 D 生成训练评分矩阵 R_d 。

第五步 根据 2.1 节中的第五步中的公式计算训练评分矩阵 R_d 中用户 u 的所有已评分项与待预测的项目 i 的差异值, 并生成项目差异性矩阵 D_k^* 。

第六步 根据 2.2 节中第四步中的公式计算用户 u 对项目 i 的预测评分值 $p_{u,i}$ 。

3 实验仿真与分析

3.1 实验数据

实验采用的数据为美国明苏达大学开发并公布的 MovieLens 十万条数据集, 该数据集中每个用户至少对 20 部电影进行了评分。用户信息结构为(年龄, 性别, 职业, 住址)。数据集分 5 次从 10^6 条记录中选取了 80% 作为训练集, 20% 作为测试集, 形成了 U1base、U2base、U3base、U4base、U5base 和 U1test、U2test、U3test、U4test、U5test。

3.2 实验过程

我们选取 U1base 和 U1test 数据集中的数据生成项目差异性矩阵, 然后计算用户预测评分和 U1test 中的真实评分的 MAE 值, 进而得到所有用户在每个项目类别下的个性化情境。接着利用从 U1base 和 U1Test 得到的用户在每个项目类别下的个性化情境对 U2test、U3test、U4test、U5test 中的项目预测评分, 计算它们和真实评分的 MAE 值以验证算法的精准性。例如在 U1base 和 U1test 数据集基础上计算得到的用户在每个项目类别下的个性化情境(见表 2)。

表 2 项目类别下的个性化情境

	Action	Adventure	Animation	...	Unknown
User1	Sex	Age	Age	...	Age
⋮	⋮	⋮	⋮	⋮	⋮
User501	Occupation	Age	Age	...	Age
⋮	⋮	⋮	⋮	⋮	⋮
User943	Age	Age	Age	...	Age

3.3 预测结果的统计分析

把算法和经典的 Slope One^[5] 进行比较, 具体比较结果如图 1 所示。实验随机采用的 U2 数据集, 即以 U2base 中的评分数据, 通过算法预测 U2test 中项目的评分, 然后将预测评分和真实评分进行比对。图 1 中的横坐标为项目评分次数, 表示该项目被用户评分次数, 例如 0-11 表示项目有 0-11 个用户对该项目进行评分。纵坐标为 MEA 值, 表示该项目的用户预测评分与真实评分的比较值, 该值越小, 推荐的准确率越高。左侧柱形表示 Slope One 算法预测的结果, 右侧柱形表示本文算法预测的结果。

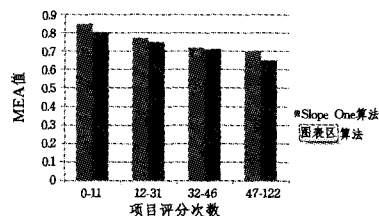


图 1 实验结果图

从图 1 中可以看出:

(1) 随着用户评分的次数的增加, 无论是 Slope One 算法还是本文算法, 预测的准确性都在提高, 这是因为随着用户评分次数的增加, 项目之间的差异性越来越接近真实值, 所以预测的效果好。

(2) 从 4 个区间段上来看, 本文算法都比 Slope One 算法的预测准确性要高。

结束语 本文结合了用户个性化情境和项目类别在传统协同过滤推荐算法中的优势。先对项目分类, 然后在每个项目类别下引入用户个性化情境, 在个性化情境和项目类别之间建立起对应关系, 消除了将一种个性化情境用于所有项目的盲目性, 加强了项目之间的联系性。在对目标项目预测评分时, 只需要参照同类别下的其他项目的评分数据和与他们之间的差异值, 这不仅提高了推荐的准确性, 也降低了算法的时间复杂度。

参考文献

- [1] 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
- [2] Sarwar B, Karyp I G, Konstan T J, et al. Item 2based collaborative filtering recommendation algorithms [C]// Proceedings of the 10th International Conference on World Wide Web. Hong Kong, 2001;285-295
- [3] 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
- [4] 高旻, 吴中福. 基于个性化情境和项目的协同推荐研究[J]. 东南大学学报, 2009, 39(增刊):1001-0505

(下转第 194 页)

用户和项目数据结合,利用 Slope One 算法计算出测试集中项目的预测分数;

2)按照 3.2 节中的步骤 1)一步骤 2)计算出训练集和测试集中全部项目的语义相似度矩阵 $[\mu_{ij}]_n$;

3)根据式(7)计算出测试集中项目基于项目语义相似的预测分数;

4)由式(8)计算出测试集中项目的最终预测分数。

4 实验结果及分析

4.1 实验数据集及检验标准

本文针对所设计的算法在 Hadoop 平台上进行了测试。实验数据采用了 MovieLens 提供的标准数据集,它包含了 943 个用户对 1682 部电影的 100000 次评分,其中在文件 movies.dat 中记录了这 1682 部电影的名称及类型^[11]。为了对比本文所提算法的效果,首先将这 100000 次评分按 4:1 的比例随机抽取形成 3 组训练集和测试集,另外还有 2 组数据训练集仍然随机进行抽取,但对测试集中的候选项目限制对其评过分的用户数占总用户数的百分比不超过 10%。

采用的评估预测精度验证方法是广泛使用的衡量用户推荐算法精度的方法——平均绝对误差(Mean Absolute Error, MAE):

$$MAE(f) = \frac{1}{R_{test}} \sum_{r_{ui} \in R_{test}} |f(u, i) - r_{ui}| \quad (9)$$

式中, $f(u, i)$ 是用户 u 对项目 i 的预测评分, r_{ui} 是测试集 R_{test} 中项目的实际评分。通过计算项目的实际评分与预测评分之间的平均误差来度量算法推荐的精度,MAE 值越小,算法性能就越好。

4.2 实验结果的分析

本文将修改后的 Slope One 算法与原算法进行对比,实验结果如图 1 所示。

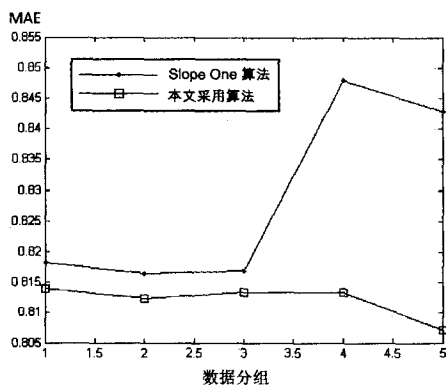


图 1 分组实验结果

图 1 中横坐标表示用于实验的数据分组,纵坐标表示每组数据分别用两种算法求的 MAE 值。

从图上来看,针对前 3 组随机抽取的数据,本文改进的算

法 MAE 值比 Slope One 算法有一定程度的减小。相对于文献[5]中提出的一种改进的加权 Slope One 算法,本文提出的算法 MAE 值减小的幅度更大。而特意选取评分较少的项目构成测试集的第 4、5 组数据,两种算法的 MAE 值相差较大。因此,实验结果表明,Slope One 算法的精度受对候选项目评过分的用户数量影响较大,而本文采用的算法在不降低原来算法性能的前提下,减小了这种影响,从而在一定程度上提高了预测的精度。

结束语 本文分析了 Slope One 算法在对预测项目评分用户数目较少的情况下可能面临的预测精度降低的问题。因此,本文利用信息检索中常用的计算概念语义相似度的方法来度量项目之间的相似程度,对同一用户已有的评分数据做出预测,充分考虑了用户自身的喜好。实验结果也表明,将基于项目语义相似度的模型与 Slope One 算法结合后可以改善原算法预测评分的质量。

参考文献

- [1] Mahmood T, Ricci F. Improving recommender systems with adaptive conversational strategies[C]// HT '09 20th ACM Conference on Hypertext and Hypermedia. New York, USA: ACM Press, 2009: 73-82
- [2] 刘浩森,徐从富,何俊. 基于模糊聚类的歌曲智能推荐方法研究[J]. 计算机工程与设计, 2009, 30(10), 2423-2427
- [3] 刘志昆,王卫平. 基于精确序列格式的网页个性化推荐[J]. 计算机系统应用, 2006, (5), 32-35
- [4] 周涛,李华. 基于用户情景的协调过滤推荐[J]. 计算机应用, 2010, 30(4): 1076-1082
- [5] Lemire D, Maclachlan A. Slope One Predictors for Online Rating Based Collaborative Filtering[C]// SIAM Data Mining (SDM'05). 2005: 21-23
- [6] Miller G A. WordNet A Lexical Database for English[C]// Communications of the ACM. New York, USA: ACM Press, 1995: 39-41
- [7] Lin De-kang. An Information-Theoretic Definition of Similarity [C]// Proceedings of the 15th International Conf. on Machine Learning Morgan Kaufmann. San Francisco, CA, 1998: 296-304
- [8] Budanitsky A, Hirst G. Evaluating WordNet-based Measures of Lexical Semantic Relatedness [J]. Association for Computational Linguistics, 2005, 32: 16-23
- [9] Resnik. Using information content to evaluate semantic similarity[C]// Proc. 14th International Joint Conference on Artificial Intelligence. 1995: 448-453
- [10] Soman K P, Diwakar S, Ajar V. 数据挖掘基础教程[M]. 范明, 牛常勇, 译. 北京: 机械工业出版社, 2009
- [11] MovieLens Data Sets. 2011-5-8 [EB/OL]. <http://www.grouplens.org>

(上接第 177 页)

- [5] Lemire D, Maclachlan A. Slope one predictors for online rating-based collaborative filtering [C]// Proceedings of the SIAM Data Mining Conference. Newport Beach, CA, USA: Society for Industrial Mathematics, 2005: 21-25
- [6] Adomavicius G, Sankaranarayanan R, Sen S, et al. Incorporating

contextual information in recommender systems using a multi dimensional approach [J]. ACM Transactions on Information Systems (TOIS), 2005: 23(1): 103-145

- [7] 戴亚娥, 龚松杰. 个性化服务中基于模糊聚类的协同过滤推荐 [J]. 计算机工程与科学, 2009, 31(4): 1007