华南理工大学学报(自然科学版)
第41卷 第9期 Journal of South China University of Technology (Natural Science Edition) Vol. 41 No. 9
2013年9月 September 2013

文章编号: 1000-565X(2013)09-0065-06

协同标签系统中基于标签组合效应的推荐算法*

蔡毅 刘宇 张广怡 陈俊挺 闵华清

(华南理工大学 软件学院,广东 广州 510006)

摘 要: 协同标签系统中现有的用户建模方法将用户视为标签向量,并假设向量中的标签均是用户感兴趣的,且只能分别计算单个标签之间的匹配程度,忽略了多个标签作为一个整体对用户兴趣产生的影响. 为此,文中提出了一种基于标签组合效应的推荐算法(TGER). 该算法利用用户对资源的评分筛选出对用户兴趣有重要影响的标签组合,通过高维标签组合优先匹配的方法计算用户与资源之间的相关度. 在MovieLens数据集上的实验结果显示,TGER算法能明显地提高推荐的质量.

关键词: 协同标签系统; 标签组合效应; 用户建模; 推荐算法

中图分类号: TP311 doi: 10.3969/j.issn.1000-565X.2013.09.011

随着互联网技术的迅速发展,推荐系统已广泛应用于电子商务与社交网络平台[1]. 在这些电子商务平台中存在着大量的用户个性化信息. 对这些信息进行分析并将结果运用到这些平台中无疑会给用户带来更好的个性化服务.

协同标签系统允许用户使用标签来描述自己评价过的资源(如电影、网页、书籍、音乐等). 采用标签来标注资源有助于资源的组织、检索和共享. 当前很多用于协同标签系统的推荐方法通过分析用户过去的行为来对用户进行建模,以分析用户的喜好. 这些方法主要采用向量空间模型对用户进行建模,即一个用户由一个包含该用户标注过的所有标签的向量来表示. 在这种用户模型中,用户对一个资源的喜好程度正比于其对该资源任意一个特征的喜好程度,故忽略了特征间的组合效应. 但特征间的组合有可能会改变特征原有的属性,即用户的喜好程度.

为此,文中提出了一种基于标签组合效应的推荐算法(TGER),并将其应用于协同标签系统中的资源推荐. 首先,在原有的向量空间模型的基础上,通过分析用户对商品的评分与对该商品标注的标签的关系获取那些造成用户对资源评分具有稳定影响的标签集合,计算标签组合聚集强度,即标签组合影响该用户对资源评分的稳定程度. 由于标签组合可能含有多个标签(称为多维标签组合),因此需要通过对较低维标签组合的聚集强度进行过滤以得到较高维的标签组合,即稳定影响用户对资源喜好程度的标签组合,并建立一个标签组合效应矩阵. 然后计算待推荐商品在矩阵中的最高维标签组合对用户评分产生影响的权值,最后整合这些权值可得出用户对目标商品的最终预测评分.

1 背景与相关工作

1.1 推荐方法

当前的推荐方法主要有基于内容的推荐方法、协同过滤推荐方法及混合推荐方法. 基于内容的推荐方法通过将目标商品与活跃用户曾经评价过的商

收稿日期: 2013-03-06

* 基金项目: 国家自然科学基金资助项目(61300137);广东省自然科学基金资助项目(S2011040002222);广东省优秀青年创新人才培育项目(LYM11019);华南理工大学中央高校基本科研业务费专项资金资助项目(2012ZM0077);国家大学生创业创新训练计划项目(201210561106,201210561108)

作者简介: 蔡毅(1980-),男,博士,副教授,主要从事数据挖掘、信息检索研究. E-mail: ycai@scut.edu.cn

品进行相似度比较,将与活跃用户喜欢的商品类似的目标商品推荐给用户[2]. 协同过滤推荐方法分为基于项目的和基于用户的两类[3]. 基于项目的协同过滤推荐方法与基于内容的推荐方法的最大区别在于相似度的计算策略,前者通过商品或用户的社会关系来计算二者的相似度,而后者则通过比较商品间"内容"的相似程度得出. 为避免这两种方法的缺点,一些学者将这两种方法结合起来,提出了混合推荐方法. 文献[4]中对混合推荐方法做了详细的分类. 文献[5-7]中探讨了一些混合推荐方法及其实现.

与基于内容的推荐方法相比,协同过滤推荐方法在很大程度上因不需要依赖于商品的内容数据而被更为广泛地运用到社交网络与电子商务等领域的推荐系统中. 学术界对协同过滤推荐方法进行了广泛而深入的研究,许多改进的协同过滤推荐方法也迅速发展起来. 文献[8]中探讨了协同过滤推荐方法的性能与效率. 在众多的协同过滤推荐方法中,基于矩阵分解的协同过滤推荐方法[9]较为流行.

1.2 协同标签系统中的用户模型

在协同标签系统中,许多推荐方法通过用户历史行为和信息对用户进行建模,其中向量空间模型是一种比较常见的用户模型.

在向量空间模型中,用户被映射到标签向量的一个空间. 这个向量空间中每个标签的权重反映了用户对该标签感兴趣的程度,权重值在[0,1]区间上. 如果权重值越接近1,意味着用户对此标签表现出越多的兴趣. 向量空间模型中标签权重的计算方法有词频(TF)及词频-逆用户频率(TF-IUF)[10]. 通常采用TF和词频-逆资源频率(TF-IRF)方法来计算每个标签在资源向量中的权值.

文献[11]中提出的标签权值计算指标归一化词频(NTF)反映了用户使用该标签的可能性或概率,故可以更恰当地反映用户对标签的感兴趣程度.

由于标签是一种文本数据,故其包含了特定的用户语义. 在向量空间模型中,如果将向量中的元素视为商品或用户的特征,则可以得到特征向量模型. 文献[12-15]中探讨了这一模型的几种实现方式.

然而,目前所使用的协同标签系统的用户建模方法大都只分别存储单个标签的权值,因而无法体现标签组合对原来标签权值可能造成的改变,但此改变可以使用户更喜欢或不喜欢某个资源. 例如,艾米是美国人,她喜爱包含"辣"或"鸡"元素的菜(如麻婆豆腐、白切鸡),但不喜欢"辣子鸡". 另一方面,艾米讨厌大蒜和菠菜,但对蒜蓉菠菜情有独钟.

针对上述问题,文献[16]中采用标签协同效应矩阵来储存每个用户使用过的任意两个标签间的组合效应特征值. 受其启发,文中对二维标签协同效应矩阵进行扩展,使该矩阵储存的不仅仅是两个标签间的效应特征值,还包括任意多个标签间的效应特征值,以使推荐结果更为准确.

2 TGER算法的建立

2.1 标签组合效应

标签组合效应是指不同的标签组合对用户喜好有不同程度的影响. 而用户喜好可以理解为用户对商品的感兴趣程度,具体可以表现为对商品的评分.

标签组合是一个或多个标签的集合,即 $f^k=\{t_1,t_2,\cdots,t_k\}$,$k$ 为 f 的维度.

为了量化标签组合效应,需要赋予其两个特征属性:标签聚集强度和标签组合权值.

首先作如下假设:若用户 i 对包含标签组合的商品的评分波动很大,则该标签组合对用户 i 的喜好的影响不明显;否则,其对用户 i 的喜好有较为强烈的影响.

用标签聚集强度 $I_i^{f^k}$ 表示标签组合 f^k 影响用户 i 喜好的稳定程度,其值域为[0,1]. 当 $I_i^{f^k}$ 趋于1时,表明 f^k 越稳定地反映用户的特定喜好;当 $I_i^{f^k}$ 趋于0时,表明 f^k 对用户喜好的影响越不稳定.

标签聚集强度并不能直接体现用户对一组标签的喜恶程度,其反映的是这组标签体现用户喜好的能力,并作为筛选更高维(指标签组合包含的标签数目)标签组合的依据. 例如,李四观看了五部电影并作如表1所示的评价.

表1 李四的电影评价记录

Table 1 Movie evaluation records for Li Si

电影名称	标签集	评分[0.5,5.0]
2012	科幻,灾难	5.0
源代码	科幻,灾难,惊悚	1.0
盗梦空间	科幻,动作	3.0
阿凡达	科幻,动作	3.0
变形金刚	科幻,动作	3.5

从表1可知:拥有"科幻"标签的电影并没有获得李四稳定的评分,即"科幻"并不能很好地体现李四的喜好;在李四对"源代码"的评价中,原本出现

在 5.0 评分的电影“2012”中的标签组合{科幻,灾难}却因多了新的标签“惊悚”而得到 1.0 的差评;李四对使用{科幻,动作}标签组合进行描述的电影的评分均为 3.0 左右,故该标签组合很有可能体现李四对电影的“一般”喜好程度.

2.2　基于标签聚集强度的用户模型

为克服传统用户模型无法准确体现用户对某标签组合的喜好程度的问题,理论上新的用户模型必须存储每一个能独立对用户喜好造成影响的标签组合的权值. 然而,试图直接穷举所有可能的标签组合是不实际的. 可取的方法就是存储一些标签组合的聚集强度,以实现低维过滤的效果,即对较低维的标签组合进行筛选,以过滤掉不符合条件的较高维的标签组合.

基于标签聚集强度的用户模型表示为一个 n 维标签组合矩阵 $\boldsymbol{M}_{i,n}$,矩阵中的每个单元存储用户 i 所使用过的标签组合对的标签聚集强度,其中 f_m^n 的上标 n 为 f 包含的标签数,下标 m 为索引.

$$\boldsymbol{M}_{i,n}=\begin{array}{c} \\ f_1^1 \\ f_2^1 \\ \vdots \\ f_m^n \end{array}\begin{array}{c} \begin{array}{cccc} f_1^1 & f_2^1 & \cdots & f_m^n \end{array} \\ \begin{bmatrix} I_i^{f_1^1,f_1^1} & I_i^{f_1^1,f_2^1} & \cdots & I_i^{f_1^1,f_m^n} \\ I_i^{f_2^1,f_1^1} & I_i^{f_2^1,f_2^1} & \cdots & I_i^{f_2^1,f_m^n} \\ \vdots & \vdots & & \vdots \\ I_i^{f_m^n,f_1^1} & I_i^{f_m^n,f_2^1} & \cdots & I_i^{f_m^n,f_m^n} \end{bmatrix} \end{array}.$$

文中基于标签聚集强度的用户模型的构建是通过过滤较低维的标签组合寻找较高维的标签组合,并以此迭代更新整个标签组合效应矩阵. 为此,首先建立一维标签组合效应矩阵 $\boldsymbol{M}_{i,1}$,每个单元存储两个标签间的聚集强度.

$$\boldsymbol{M}_{i,1}=\begin{array}{c} \\ f_1^1 \\ f_2^1 \\ \vdots \\ f_m^1 \end{array}\begin{array}{c} \begin{array}{cccc} f_1^1 & f_2^1 & \cdots & f_m^1 \end{array} \\ \begin{bmatrix} I_i^{f_1^1,f_1^1} & I_i^{f_1^1,f_2^1} & \cdots & I_i^{f_1^1,f_m^1} \\ I_i^{f_2^1,f_1^1} & I_i^{f_2^1,f_2^1} & \cdots & I_i^{f_2^1,f_m^1} \\ \vdots & \vdots & & \vdots \\ I_i^{f_m^1,f_1^1} & I_i^{f_m^1,f_2^1} & \cdots & I_i^{f_m^1,f_m^1} \end{bmatrix} \end{array}.$$

提取矩阵 $\boldsymbol{M}_{i,1}$ 中聚集强度大于或等于 α 的二维标签组合(其中 α 为标签聚集强度阈值,只有当标签组合的聚集强度大于或等于 α 时,其组合效应才不可忽略),然后构建 $\boldsymbol{M}_{i,2}$,$\boldsymbol{M}_{i,2}$ 增加的行和列是从 $\boldsymbol{M}_{i,1}$ 中提取出来的二维标签组合. 同理,通过 $\boldsymbol{M}_{i,2}$ 可以迭代构建 $\boldsymbol{M}_{i,4}$,依此类推,直到新矩阵与其前一矩阵扩展出来的部分相比没有任何单元的值大于或等于 α 时,迭代停止. 最后获得一个矩阵 $\boldsymbol{M}_{i,n}$,该矩阵单元存储所有不可忽略组合效应的标签组合的聚集强度,这些标签组合包含的标签数目范围为 $[2,n^2]$.

2.3　标签聚集强度计算方法

标签聚集强度作为反映标签组合影响用户喜好的稳定程度的特征量,理论上应当满足如下约束条件: 如果一个标签组合的所有子标签组合的聚集强度均大于常数 α,则该标签组合的聚集强度也大于 α,即

$$\forall f^k\subseteq f^{k+1},I_i^{f^k}>\alpha\Rightarrow I_i^{f^{k+1}}>\alpha.$$

该约束可以确保不可忽略组合效应的较高维标签组合不会因为其不满足条件的较低维子标签组合被过滤而被过滤,另一方面,对于所有可以忽略组合效应的高维标签组合在其低维子标签组合被过滤后随即被过滤. 满足此约束条件的计算方法不仅可以明显地提高基于标签组合效应的推荐算法的推荐性能,而且使得高维标签组合效应矩阵的建立成为可能.

考虑到算法时间复杂度的因素,文中给出一种朴素的标签聚集强度计算方法,即

$$I_i^f=\frac{|i(f)|}{\sum_{j\in i(f)}(r_{i,j}-\bar{r})^2} \tag{1}$$

式中,$i(f)$ 为用户 i 使用标签组合 f 标注过的商品集合,$|i(f)|$ 为用户 i 使用标签组合 f 标注商品的总次数,$r_{i,j}$ 为用户 i 对商品 j 的实际评分,$\bar{r}$ 为用户 i 对标签组合 f 的平均评分.

该方法采用方差特征值的倒数,将其映射到 $[0,1]$. 基于式(1),文中提出了一种标签组合效应矩阵生成算法,具体的算法描述如下:

```
{ 输入: 用户 i 评价过的商品的记录
  输出: 用户 i 的标签组合效应矩阵 M_i
  for 用户 i 使用过的标签 t_x do
      标签容器 f.add(t_x)
      当前标签组合容器λ.add(f)
  end for
  while λ≠null do
      临时标签组合容器 T ← ∅
      for λ 中的每一个元素 f_a do
          for λ 中的每一个元素 f_b do
              f ← f_a + f_b
```

$$I_i^f \leftarrow \frac{|i(f)|}{\sum_{j\in i(f)}(r_{i,j}-\bar{r})^2}$$

$$M_i^f \leftarrow \frac{2\arctan I_i^f + \pi}{2\pi}$$

if $M_i^f \geqslant \alpha$ then

T. add(f)

end if

end for

end for

λ. clear( )

$\lambda \leftarrow T$

end while}

由式(1)可知,遍历一次用户 i 标注过的标签组合 f 的商品列表,即可求出 I_i^f,通常这个列表的长度不大于 10. 因此,该算法的运行时间主要消耗在 while 循环中. 假设标签组合效应矩阵包含所有可能的标签组合,即 $\alpha = 0$,那么采用该算法构建 n 维标签组合效应矩阵的时间复杂度上界为 $O(q^n)$,其中 q 为用户 i 使用过的标签数目. 在实际情况中,通过调节阈值 α 可以过滤掉大量的标签组合,因此时间复杂度远小于 $O(q^n)$. 标签组合效应矩阵生成算法的空间消耗主要在于存储标签组合效应矩阵,其空间复杂度上界为 $O((ls)^n)$,其中 l 为最大标签组合的长度,s 为矩阵中标签组合的数目.

2.4 TGER 算法描述

标签组合权值体现用户对该标签组合的喜好程度,其主要应用于基于标签组合效应的推荐过程.

用户对标签组合的喜好程度由其组合权值体现,权值越大表示用户对出现该标签组合的商品越感兴趣. 标签组合权值的计算方法如下:

$$\hat{r}_{i,j} = \frac{1}{|F(i,j)|}\sum_{f\in F(i,j)} w_i^f \tag{2}$$

式中,$\hat{r}_{i,j}$ 为 TGER 算法预测用户 i 对商品 j 的评分,$F(i,j)$ 为用户 i 标签组合容器与商品 j 均包含的标签组合集合,$|F(i,j)|$ 为 $F(i,j)$ 的长度,w_i^f 为标签组合 f 的权值. 该方法属于高维标签组合优先匹配计算方法,即一旦找到更高维度的标签组合,就以该维度的标签组合权值作为推荐的依据,而不再考虑较低维度的标签组合权值对推荐的影响.

由于 w_i^f 依赖于标签组合 f 出现的总次数以及每次出现时用户 i 对商品 j 的评分,故文中给出 w_i^f 的两种计算方法:

(1) 组合效应传递法. 此方法最初是为了解决数据集的稀疏问题而提出的,其基本思想是:$\{t_1,t_2\}$、$\{t_1,t_3\}$ 和 $\{t_2,t_3\}$ 均出现过,但 $\{t_1,t_2,t_3\}$ 没有. 然而,因为每两个标签组合的交集都非空,所以先假定 $\{t_1,t_2,t_3\}$ 很可能会出现,并且根据 $\{t_1,t_2\}$、$\{t_1,t_3\}$ 和 $\{t_2,t_3\}$ 的聚集强度去计算 $\{t_1,t_2,t_3\}$ 的加权出现次数,其中最简洁的一种方法是取其平均数.

(2) 组合效应加权法. 此方法只在 $\{t_1,t_2,t_3\}$ 共同出现时才去计算其组合权值的增量,其中一种计算方法为

$$w_i^f = \frac{1}{|i(f)|}\sum_{j\in i(f)}\frac{1}{C_c^k} r_{i,j} \tag{3}$$

式中,c 为用户 i 标注商品所使用的标签总数. 注意到用户使用 c 个标签去描述一个商品时,其评分取决于这些标签的整体影响,此时标签组合的任意一个子集对最终评分均只能起到部分的影响,故加上 $\frac{1}{C_c^k}$ 作为部分匹配的罚项. 采用式(3)的基于标签组合效应的推荐算法描述如下:

{输入:用户 i 的标签组合容器 λ_i,商品 j 的标签集 f_j

输出:预测评分 $\hat{r}_{i,j}$

临时标签组合容器 $T \leftarrow \varnothing$

k = getMaxDimension(λ_i, f_j)

for λ_i 中的每一个 k 维标签组合 f_x^k do

if $f_x^k \subseteq f_j$ then

T. add(f_x^k)

end if

end for

$$\hat{r}_{i,j} = \frac{1}{|T|}\sum_{f\in T} w_i^f$$

}

该算法的运行时间主要消耗在构建临时标签组合容器 T 中,维护一张全局的标签索引的时间复杂度为 $O(k)$,因此该算法的时间复杂度上界为 $O(gl)$,其中 g 为 λ_i 的长度.

3 实验与结果分析

为检测文中所提算法的效率,采用 MovieLens 数据集(该数据集已被广泛应用于推荐系统的相关实验中)进行实验,抽取了其中同时包含评分和标签的数据项组成一个新的数据集,该数据集包含 44 805 条记录,每条记录为一个四元组〈用户 ID,电影 ID,标签,评分〉,由 2 025 个用户对 4 796 部电影

的评价构成. 每个用户至少评价了一部电影,每部电影至少使用一个标签,每个标签是一个英文单词或一句简短的话. 分数的域值为 [0.5,5.0],间隔为 0.5. 将每个用户评价过的电影的 80% 作为训练集,剩下的 20% 作为测试集,用以测试基于标签组合效应的推荐算法的效率.

实验采用绝对平均误差(MAE)来评估目标算法的效率,其定义如下:

$$\mathrm{MAE}=\frac{1}{N}\sum_{i=1}^{N}|r_{i,j}-\hat{r}_{i,j}| \tag{4}$$

式中,N 为被预测的商品总数.

本实验采用经典的 CB 算法和 MovieLens 数据集上广泛使用的 MF 算法作为基准参照算法. 由于数据集自身的稀疏性,当标签组合过滤到三维以后,四维标签集合极为稀少,其对推荐效果的提高极为微弱,因而本次实验只对一维、二维和三维的 TGER 算法进行测试.

为探讨基于标签聚集强度的用户模型的性能,以及验证 TGER 算法的效率,文中的实验步骤如下: ①对不同维度的标签聚集强度过滤方法进行对比,以探讨用户模型的性能随存储的标签组合维度的增加而变化的趋势; ②将 TGER 算法与参照算法进行对比,考察基于标签聚集强度的用户模型应用在推荐系统中的效率.

实验观察到数据集中的标签组合的聚集强度主要分布在 [0.80,0.99],由于篇幅有限,文中只给出标签聚集强度阈值 α 取 0.85、0.94 和 0.97 时的实验结果.

为考察 TGER 算法对小规模及不同区域的测试对象的 MAE 性能,将整个测试集包含的 2 025 个用户随机地划分成 5 个测试组,分别使用一、二、三维 TGER 算法对这 5 个测试组进行测试,结果如图 1 所示. 可以看出,当过滤维数逐渐提高时,TGER 算法的 MAE 性能越来越好.

从图 1 还可以发现,针对实验所用数据集,标签聚集强度阈值对 TGER 算法性能的影响并不明显. 理想的情况应当是: 在不降低 TGER 算法性能的前提下,最大程度地过滤掉标签组合.

在整个测试集范围内进行测试并对 TGER 与 MF、CB 算法的 MAE 性能进行对比,当 $\alpha=0.97$ 时,一维 TGER、二维 TGER、三维 TGER、MF 和 CB 算法的 MAE 分别为 0.56、0.52、0.50、0.62 和 0.89. 三维 TGER 算法的 MAE 性能相对于 MF 与 CB 算法分别提高了 19% 和 44%.

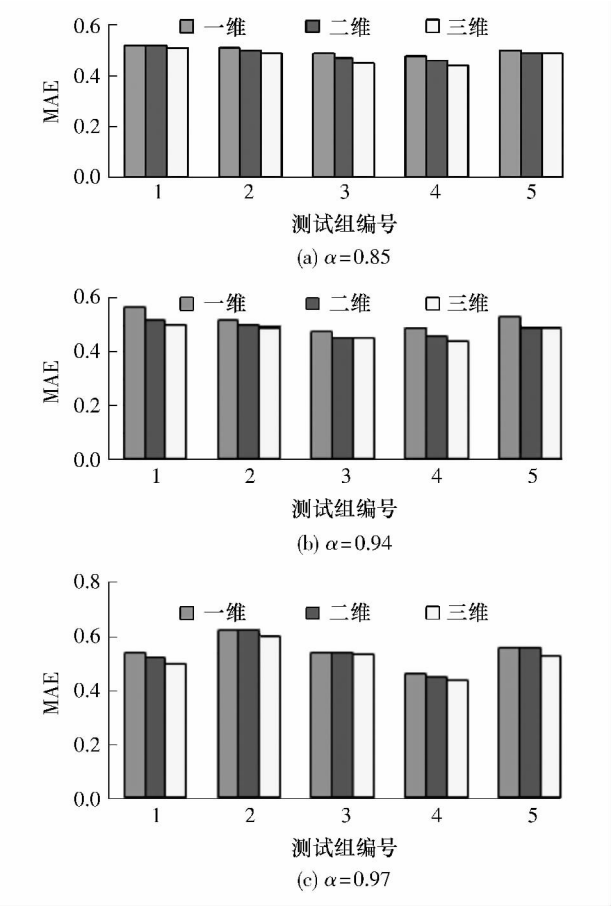

图 1　一维、二维和三维 TGER 算法的 MAE 对比

Fig. 1　Comparison of MAE among one-, two- and three-dimension TGER algorithm

4　结语

在分析现有推荐方法忽略标签组合效应所带来的局限性后,文中探讨并定义了标签组合效应,提出了一种新的基于标签组合效应矩阵和用户对资源评分的用户建模方法,并基于该用户建模方法提出了一种新的推荐算法,最后采用 MovieLens 数据集对其进行实验评估,结果表明,文中提出的推荐算法能够明显地提高推荐的质量. 今后拟将该用户建模方法应用到个性化搜索中.

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 Transactions on Knowledge and Data Engineering,2005,17(6): 734-749.

[2] Lops P,de Gemmis M,Semeraro G. Content-based recommender systems: state of the art and trends [M]. Recommender Systems Handbook. New York: Springer-Verlag,2011: 73-105.

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

[4] Burke R. Hybrid recommender systems: survey and experiments [J]. User Modeling and User-Adapted Interaction, 2002, 12(4): 331-370.

[5] Melville P, Mooney R J, Nagarajan R. Content-boosted collaborative filtering for improved recommendations [C] // Proceedings of the Eighteenth National Conference on Artificial Intelligence. Menlo Park: ACM, 2002: 187-192.

[6] Ma H, King I, Lyu M R. Effective missing data prediction for collaborative filtering [C] // Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Amsterdam: ACM, 2007: 39-46.

[7] Gunawardana A, Meek C. A unified approach to building hybrid recommender systems [C] // Proceedings of the Third ACM Conference on Recommender Systems. New York: ACM, 2009: 117-124.

[8] Symeonidis P, Nanopoulos A, Papadopoulos A N, et al. Collaborative recommender systems: combining effectiveness and efficiency [J]. Expert Systems with Applications, 2008, 34(4): 2995-3013.

[9] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8): 30-37.

[10] Noll M G, Meinel C. Web search personalization via social bookmarking and tagging [M]. The Semantic Web. Berlin/Heidelberg: Springer-Verlag, 2007: 367-380.

[11] Cai Y, Li Q, Xie H, et al. Personalized resource search by tag-based user profile and resource profile [M]. Web Information Systems Engineering. Berlin/Heidelberg: Springer-Verlag, 2010: 510-523.

[12] Hofmann T. Probabilistic latent semantic indexing [C] // Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley: ACM, 1999: 50-57.

[13] Agarwal D, Merugu S. Predictive discrete latent factor models for large scale dyadic data [C] // Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose: ACM, 2007: 26-35.

[14] Agarwal D, Chen B C. Regression-based latent factor models [C] // Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris: ACM, 2009: 19-28.

[15] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis [J]. Journal of the American Society for Information Science, 1990, 41(6): 391-407.

[16] Han H, Cai Y, Shao Y, et al. Improving recommendation based on features' co-occurrence effects in collaborative tagging systems [M]. Web Technologies and Applications. Berlin/Heidelberg: Springer-Verlag, 2012: 652-659.

Tag Group Effect-Based Recommendation Algorithm for Collaborative Tagging Systems

Cai Yi Liu Yu Zhang Guang-yi Chen Jun-ting Min Hua-qing

(School of Software Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China)

Abstract: In the existing user modeling methods for collaborative tagging systems, a user is regarded as a tag-vector and it is assumed to be interested in every tag in the tag-vector. Moreover, only the matching degree of a tag with another tag is calculated, while the effects of tags as a whole on the user's preference are ignored. In order to solve these problems, this paper proposes a recommendation algorithm based on the tag-group effect, namely, TGER. This algorithm utilizes the user ratings on resources to select the tag-groups which have significant effects on the user's preference, and adopts the high-dimension tag-group first matching method to calculate the user-resource relevance. Experimental results on the MovieLens data set show that TGER can significantly improve the recommendation quality.

Key words: collaborative tagging system; tag-group effect; user modeling; recommendation algorithm