

文章编号: 1674-2974(2011)05-0083-06

## 基于同义标签分组的协同推荐<sup>\*</sup>

陈毅波<sup>1</sup>, 揭志忠<sup>1</sup>, 吴产乐<sup>1,2</sup>

(1. 武汉大学 计算机学院, 湖北 武汉 430072; 2. 国家多媒体软件工程 技术研究中心, 湖北 武汉 430072)

**摘要:** Last.FM, MovieLens 等在线服务一个重要的特征是用户能手动使用标签标注对象, 很多研究人员使用标签改进推荐质量. 但是, 标签是一个用户对对象的局部描述, 不同的用户可能使用不同的标签来描述同一个对象, 但表达同样的意义, 这导致了单纯的使用标签进行推荐所达到的效果不理想. 因此, 本文首先根据标签共生分布计算标签相似度, 然后根据相似度分组同义标签, 最后通过组标签计算用户的相似度, 将其混合到经典的协同过滤推荐算法中. 实验结果表明, 该方法能有效地改进现有的推荐算法, 达到更好的推荐效果.

**关键词:** 推荐系统; 标签分组; 协同过滤; 标签

中图分类号: TP311

文献标识码: A

## Collaborative Recommendation Based on Synonymy Tags Grouping

CHEN Yi-bo<sup>1</sup>, JIE Zhi-zhong<sup>1</sup>, WU Chan-le<sup>1,2</sup>

(1. Computer School of Wuhan Univ, Wuhan, Hubei 430072, China;

2. National Engineering Research Center for Multimedia Software, Wuhan, Hubei 430072, China)

**Abstract:** An important facet of Last.FM and MovieLens is that users manually annotate the items using so called tags. There are many researches about using tag to improve the quality of recommendation. However, tags are “local” descriptions of items given by the users, because different people use different tags for the same item, but the tags may represent the same means. In this paper, we used tag grouping method to group the tag according to the similarity of co-occurrence distributions. Based on this, we proposed an approach to group synonymy tags and fusing the relationship between users-tag with the collaboration filtering algorithms. The results of the empirical evaluation show that the approach is effectiveness in augmenting recommendation.

**Key words:** recommender systems; tag grouping; collaboration filtering; tag

随着因特网的快速发展, 接入因特网的服务器数量和基于 WWW 的 Web 数据成指数增长. 快速发展的 Internet 在同一时刻呈现给我们大量的信息, 例如: Netflix 有超过 10 万部电影, Amazon 有上千万的书籍, Del.icio.us 有超过百亿的页面集合,

如此多的信息, 即使全部浏览一遍都不可能, 更不用说找到自己喜欢的内容. 传统的搜索算法为所有用户提供同样的排序结果, 并不能根据用户的喜好提供不同的服务. 信息的爆炸使得信息的使用效率反而降低, 这种现象被称之为“信息过载”. 在这种背景

<sup>\*</sup> 收稿日期: 2010-11-16

基金项目: 国家自然科学基金 No. 60672051, 中央高校基本科研业务费专项资金(3105005), 武汉市科学技术局科技计划项目(201010621209)

作者简介: 陈毅波(1982-), 男, 湖南长沙人, 武汉大学博士生

† 通讯联系人, E-mail: ybchen@whu.edu.cn

下,个性化推荐,包括个性化搜索被认为是解决信息过载最有效的工具之一。

一方面,个性化推荐是一个决策过程,提供一个用户可能会选择的对象的集合。这些对象涵盖范围从个人主页到各种商品,从广告到新闻。随着 Web 提供内容的多样化,站点也需要越来越复杂的方法为用户提供相关的内容。因此,个性化推荐在提供以用户为中心的内容和体验时扮演了一个至关重要的角色,在吸引客户和保持用户的忠诚度上起到了关键性的作用。个性化推荐服务的本质是使用机器代替用户评价陌生产品,这些产品包括书籍、电影、CD、WEB 等等,它是一个从已知到未知的过程<sup>[1]</sup>。

另一方面,用户独自学习每一个可能选择的花费非常高,并且针对不同的应用环境,一个用户可能是产品购买者,也可能是一个信息的查找者。因此,推荐作为一个社会过程在很多的应用中扮演了一个重要的角色<sup>[2]</sup>。

一个推荐系统包括三个部分:收集用户信息的行为记录模块,用户喜好分析模块和推荐算法模块,其中,推荐算法模块是推荐系统中的核心模块<sup>[3]</sup>。目前,推荐算法主要包括协同过滤算法,基于内容的推荐算法,基于用户—对象的二部图推荐算法和混合推荐算法。

协同过滤是目前最成功的推荐方法,协同过滤聚集相似用户在系统中的体验生成个性化的推荐,其最为关键的一个方面是根据用户的喜好判断用户之间的相似度<sup>[4]</sup>。协同过滤被应用于多个领域,其中,Grundy 被认为是第一个协同过滤系统,该系统能构建用户喜好模型,并根据该模型为用户推荐相关的书籍<sup>[5]</sup>。Tapestry 邮件处理系统,通过人工来处理用户之间的相似度,因此,用户越多,其推荐准确率越低<sup>[6]</sup>。GroupLens 系统构建用户的信息组,组中的用户可能发布自己的信息,并且与其它用户进行协同过滤<sup>[7]</sup>。Ringo 系统利用社会信息过滤方法为用户提供音乐推荐<sup>[8]</sup>。典型的协同过滤推荐系统还有 Amazon.com, Jester, Phoaks 等。

为了处理协同过滤问题,很多算法被提出,这些算法可分为两类:基于记忆的算法和基于模型的算法<sup>[9]</sup>。基于记忆的算法首先根据产品的评分,从训练集中找出前用户的相似用户,然后综合相似用户的评分来预测当前用户对未评分商品的评分。两个最为经典的相似度计算方法是 Pearson 相关系数和向量余弦夹角。在此之上也出现了很多改进型方法,例如,缺席投票、实例引申、加权优势预测等等。另外,

在文献[10]中,作者利用产品列表中的先后顺序计算用户之间的相似度,在计算用户相似度时,产品排序越高,其所据有的权重值就越大。在文献[11]中,作者提出了使用用户的行为信息构建用户的兴趣点,利用兴趣点计算用户的相似度<sup>[1]</sup>。基于模型的算法收集用户的评分数据进行学习并推理用户的行为模型,预测产品的评分。基于模型的协同过滤与基于记忆的协同过滤的不同之处在于基于模型的方法并不是基于一些启发式规则进行预测,而是基于数据的应用统计方法和机器学习方法得到一个预测模型。在文献[9]中,作者提出了两个选择概率模型:聚类模型和贝叶斯网络模型。在第一个模型中,假设用户的评分是独立的,将相似用户进行聚类,并给予该用户类一个标号;在贝叶斯网络模型中,类别的编号和模型参数能够对已存在的数据进行学习获取。其它基于模型的协同过滤系统还有最大熵模型、线性回归模型、概率相关模型等等。两类协同过滤算法中,由于基于内存的协同过滤算法在简单性、可靠性、有效性和稳定性方面的优点而被广泛使用。

协同标签系统是一个基于 Web 的系统,它允许用户上传他们的资源,并且使用任意的词汇(即标签)标注资源。当今,这些系统变得越来越流行,例如流行的 web 服务提供商 MovieLens, Flickr, del.icio.us, Last.fm 等等,允许用户使用标签标注其喜欢的对象。总之,标注正在成为一个新的趋势,使得用户可以容易地为对象添加元数据。这些附加的元数据能够用于改进搜索机制,更好地构建浏览数据或提供符合用户兴趣的个性化内容。因此,标签信息被广泛应用于基于内存的协同推荐算法中。在属性感知的协同推荐系统中,内容信息被附加于产品中,并且这类信息通常由领域专家提供。对于不同用户,产品展现出同样的属性描述。而标签由用户自己提供,因此,标签不仅仅与对象相关而且与用户相关。尽管标签和属性都是元数据,且都可以作为附加的背景知识用于改进推荐算法,但他们的处理方式却不一样。虽然在属性感知推荐算法方面拥有大量的研究,但对于集成标签到推荐算法中的探索仍然较少。大多数已存在的工作,集中于标签本身的推荐。然而,与属性不同的是,在实际应用中,不同用户可能使用不同的标签来表达对同一对象的兴趣,以至于研究人员不能充分利用标签的隐信息来改进推荐质量。

本文提出了一种集成标签到推荐系统的方法,首先对同义标签进行分组,然后扩展用户—标签矩阵,最后混合用户—标签关系至协同过滤推荐算法

中. 本文的贡献如下: 1) 提出了通用的标签分组方法; 2) 通过标签分组减少推荐过程中同义标签的计算, 从而提高了推荐算法的性能; 3) 提出了一种自适应的混合机制, 该机制可以捕获用户、产品和标签之间的三维关系.

1 研究现状

目前使用标签信息进行推荐的研究大多关注推荐标签本身, 以帮助用户更好地进行对象标注, 而利用标签信息来做产品推荐的研究相对较少. 在文献 [12] 中, 作者提出将多维矩阵划分为多个 2 维矩阵, 使得传统的推荐系统算法能够直接应用. 在文献 [13] 中, 作者提出了一个通用的方法, 该方法将 3 维矩阵降解为三个 2 维矩阵, 然后使用混合方法重新构建他们之间的联系.

为了避免 3 维矩阵分割带来隐信息丢失, 在文献 [14] 中, 作者使用高阶奇异值分解方法来处理矩阵的所有维度, 该方法揭示了对象之间的潜在关系. 在文献 [15] 中, 作者提出了一种基于用户—对象—标签三方图的集中扩散推荐算法. 该方法首先考虑一个  $G(U, I, E)$  二部图, 其中  $U$  是用户集,  $I$  是产品集,  $E$  是连接用户和产品的边的集合. 假设一种资源被初始分配于对象之上, 每个对象将资源平均分配给所有的邻居用户, 然后每个用户再将其分配到的资源平均分配给所有与之关联的对象. 最后, 作者将该方法扩展至了三方图中.

然而, 上述方法忽略了一个重要的问题, 即同义标签. 该问题将导致用户/产品之间的相似度计算出现较大的偏差. 问题如图 1(a) 所示, 假设  $tag1$  和  $tag2$  是同义标签,  $tag3$  和  $tag4$  是同义标签, 通过 Pearson 相关系数或向量余弦夹角计算相似度, 我们仅仅能发现  $user1$  和  $user2$  或  $item1$  和  $item2$  是相似的, 而实际上,  $user1$ ,  $user2$  和  $user3$  具有相同的相似度, 这种计算方法不可避免地会出现相似度计

一标签矩阵, 如图 1(b) 所示.

2 分组同义标签

对标签进行分组的前提是找出同义标签. 目前, 测量标签之间的关系是一个活跃的研究领域, 但是该方面的文献仍然较少. 已存在的工作可以归纳为两类, 一类是通过信息检索技术分析标签的内容; 另一类是基于标签分类结构的协同过滤方法. 在文献 [16] 中, 作者提出了一个基于词网和开放目录的外部扩展语义代理来判断标签的相似性. 在文献 [17] 中, 为了使用语义相关的标签来扩展用户查询, 作者提出了标签聚类方法, 但是基于聚类方法的相似度测量在分组标签词汇时太严格. 因此, 本文提出了一种新的方法来识别和分组同词标签. 该方法使用标签的自然结构来辨别同义标签, 不需要给用户强加特定的标记策略.

协同标签系统中的三元关系可看作一个 3 维的矩阵, 如图 2 所示. 我们将 3 维矩阵划分为用户—对象、用户—标签和对象—标签三个 2 维矩阵.

定义 1  $UI_{n,m}$  (用户—对象矩阵),  $n$  是用户数,  $m$  是对象数, 单元格的值是用户对产品的评分. 该矩阵的一列表示产品的用户概貌, 包含了对象的用户分布.

定义 2  $UT_{n,k}$  (用户—标签矩阵),  $n$  是用户数,  $k$  是标签数, 单元格的值表示用户使用该标签的次数. 该矩阵的一列表示标签的用户概貌, 包含了标签的用户分布.

定义 3  $IT_{m,k}$  (对象—标签矩阵),  $m$  是对象数,  $k$  是标签数, 单元格的值表示多少用户使用某一标签标注某一产品. 该矩阵的一列表示标签的对象概貌, 包含标签的对象分布.

标签分组过程如下.

构建初始分组矩阵: 基于对象和用户的共生分布, 根据公式(1)计算标签之间的 Pearson 相似度, 得到相似度矩阵  $SimUT_{k,k}$  和  $SimIT_{k,k}$ , 并构建一个初始标签分组矩阵  $Label\_Initial\_Group = SimUT_{k,k}$ . 遍历初始矩阵中的每一行, 其元素值大于 0.5 的元素不变, 小于 0.5 的置为 0 (即根据  $SimIT_{k,k}$ , 保留相似度大于 0.5 的元素); 然后根据相似度矩阵  $SimUT_{k,k}$  过滤掉初始矩阵中相似度大于 0 的标签, 如图 2 所示.

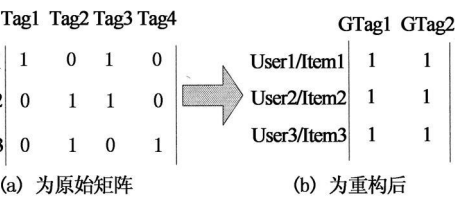


图 1 用户/对象—标签矩阵  
Fig. 1 User/object tag matrix

算偏差. 因此, 为了提高相似度计算的精确度, 需要一种方法来发现并分组同义标签, 并重构用户/产品

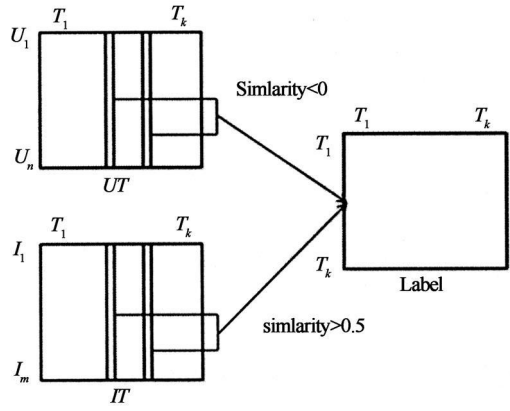


图 2 初始标签分组过程

Fig. 2 The initial grouping process of tags

$$Sim(x, y) = Pearson(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}} \quad (1)$$

构建标签组向量: 遍历初始标签矩阵  $Label\_Initial\_Group$ . 如果  $Label\_Initial\_Group(i, j) >$

0.8, 将除  $Label\_Initial\_Group(i, j)$  外, 第  $j$  行和  $j$  列的元素设为 0. 使用向量  $\vec{Tag_i\_Name} = \{label1\_name, label2\_name, \dots, labeln\_name\}$  表示初始分组矩阵中不为 0 的行, 每个向量包含该行中不为 0 的元素的列名 (即标签名), 最终形成一个分组标签向量集. 该集合大于 1 小于总的标签数.

向量去重获取标签分组: 遍历整个向量集, 从向量集中依次取出两个向量进行交集运算  $\vec{Tag_i\_Name} \cap \vec{Tag_j\_Name}$ . 通过交集运算, 得到 0 个或多个标签, 形成一个标签集  $\{tag_1, tag_2, \dots, tag_k\}$ , 在  $Label\_Initial\_Group$  矩阵, 分别查找标签集中各标签与标签名为  $Tag_i\_Name$  和  $Tag_j\_Name$  的标签相似度, 如果  $Sim(tag_k, Tag_i\_Name) > Sim(tag_k, Tag_j\_Name)$ , 则将标签向量  $\vec{Tag_j\_Name}$  中的  $tag_k$  删除, 否则删除标签向量  $\vec{Tag_i\_Name}$  中的  $tag_k$ .

通过上述三步将得到一个向量集, 其中向量数为最后的标签分组数, 每个向量中的元素即为相似标签. 算法 1 显示了标签分组的过程.

Tab. 1 The process of tag grouping

Algorithm 1: Tag Grouping Algorithm

```
Input:  $UT_{n,k}$ ,  $IT_{m,k}$ ; //  $UT_{n,k}$  is User-Tag matrix,  $IT_{m,k}$  is Item-Tag matrix
Output:  $Glabe1 = \{Glabe11, Glabe12, \dots, Glabe1n\}$ ; // Tag group vector set
Using formula(1) to compute tag similarity matrix  $SimUT_{k,k}$  based on  $UT_{n,k}$ 
Using formula(1) to compute tag similarity matrix  $SimIT_{k,k}$  based on  $IT_{m,k}$ 
 $Label\_Initial\_Group = SimUT_{k,k}$ ; //  $Label\_Initial\_Group$  represent initial grouping matrix;
Foreach ( $Label\_Initial\_Group(i, j)$  in  $Label\_Initial\_Group$ )
    If ( $(Label\_Initial\_Group(i, j) < 0.5$  or  $(Sim(i, j) > 0)$  then
         $Label\_Initial\_Group(i, j) = 0$ 
    endifor
Foreach ( $Label\_Initial\_Group(i, j)$  in  $Label\_Initial\_Group$ )
    If ( $Label\_Initial\_Group(i, j) > 0.8$ ) then
        All of elements of the  $j^{th}$  row are set to 0;
        All of the element of the  $j^{th}$  column are set to 0 except for the  $Label\_Initial\_Group(i, j)$ 
    Endif
endifor
Foreach ( $Label\_Initial\_Group(i)$  in  $Label\_Initial\_Group$ ) //  $Label\_Initial\_Group(i)$  represent all of element in the  $i^{th}$  row;
    If ( $Label\_Initial\_Group(i)$  not equals 0) then
        Using vector  $\vec{Tag_i\_Name} = \{label1\_name, label2\_name, \dots, labeln\_name\}$  to represent the nonzero element in the  $i^{th}$  row; //
         $Tag_i\_Name$  is the name of the  $i^{th}$  tag;
         $label\_name$  is the tag name;
         $Glabe1.add(\vec{Tag_i\_Name})$ ; //  $Glabe1$  is a vector set of tags;
    endifor
For ( $i=1$  to  $Glabe1.Length$ )
    For ( $j=i$  to  $Glabe1.Length$ )
         $Tag\_set = \vec{Tag_i\_Name} \cap \vec{Tag_j\_Name}$ ; //  $Tag\_set$  is a set of tags;
        Foreach ( $tag_k$  in  $Tag\_set$ )
            Queing the similarity between  $tag_k$  and  $Glabe1(i).name$ ,  $tag_k$  and  $Glabe1(j).name$  in  $Label\_Initial\_Group$  matrix;
            If  $Sim(tag_k, Glabe1(i).name) > Sim(tag_k, Glabe1(j).name)$ , then deleting  $tag_k$  in the  $Glabe1()$  and vice versa;
        endifor
    endifor
endifor
```

3 通过相似度权重混合用户—标签关系

在评分矩阵中, 我们将用户喜好看作特征向量, 每一个向量包含  $n$  个特征, 每个特征对应于一个可用的对象. 每一个特征的值为用户  $u_i$  对产品  $i_k$  的评分  $r_{i,k}$ , 当不存在相应的评分时, 其值为 0. 现在, 我们用 Pearson 相似度计算用户  $x$  和用户  $y$  之间的相似度. 根据用户评分矩阵中计算得到的用户相似度, 我们给予用户相似度权重混合用户—标签关系的详细描述.

当混合用户—标签关系和评分矩阵得到的用户相似度时, 首先需要得到基于用户—标签关系的用户相似度. 根据第三节的描述, 我们使用用户—组标签来表示用户—标签关系. 如果用户使用了一个属于标签组  $G_{a_j}$  的标签, 则  $u_{i,j}$  的值加 1, 其中  $u_{i,j}$  的初始值为 0. 基于用户—组标签矩阵, 通过 Pearson 相关系数可以计算得到用户相似度, 即  $SimUG(x, y)$ . 然后通过赋予权重值的方法混合  $SimUG(x, y)$  和  $SimUI(x, y)$ , 得到最终的用户相似度, 如式 (2) 所示:

$$Sim(x, y) = aSimUI(x, y) + (1 - a)SimUG(x, y).$$

(2)

参数用于调整  $SimUG$  和  $SimUI$  的权重, 越大, 在相似度计算中, 评分矩阵所扮演的角色越重要. 最后, 我们使用混合相似度  $Sim(x, y)$  来获取每个用户的邻居.

4 实验与分析

4.1 实验数据

Last.fm 是一个音乐社会网络站点, 该站点允许用户创建用户概貌, 并且在用户收听音乐节目的过程中不断地被丰富, 这些节目可能来自于站点本身, 也可能来自于用户个人收集的本地音乐集. 在实验中, 我们使用 Last.fm 数据集, 该数据集由 Ioannis Konstas 等研究人员收集, 其中包含了 3148 个用户, 30520 个音乐跟踪和 12565 个标签.

4.2 度量方法

本文采用信息检索领域中的标准度量方法评估我们的推荐系统. 在每一轮交叉验证过程中, 系统推荐并对一个用户可能会跟踪的音乐集合进行排序, 然后我们比较推荐列表与测试集中用户实际的跟踪

列表, 并计算其精确度 (Precision)、召回率 (Recall) 和  $F$  度量 ( $F$ -measure).

召回率: 表示推荐列表中相对于用户测试集中实际跟踪对象的命中数与用户测试集中实际跟踪的对象数之比. 该值越高, 代表系统的性能越好.

$$Recall = \frac{\text{Number of recommended tracks that match with future views}}{\text{Total number of tracks in future views}}$$

(3)

精确度: 推荐列表中相对于用户测试集中实际跟踪对象的命中数与推荐对象数之比.

$$Precision = \frac{\text{Number of recommended tracks that match with future views}}{\text{Total number of recommended tracks}}$$

(4)

$F$ -measure: 准确度和召回率之间的权重调和平均.

$$F\_measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

(5)

4.3 实验过程与结果

首先评价基于权重相似度的混合方法, 我们基于用户—对象评分矩阵使用基于用户的协同过滤系统生成一个评价基线  $CF_{UI}$ . 然后混合用户—标签关系与评分矩阵. 在混合的过程中, 使用  $a$  调整评分矩阵与用户标签关系之间的权重来计算用户相似度. 图 6 表明了  $F$ -measure 随  $a$  的变化趋势, 当  $a=0.6$  时, 其值达到了峰值, 这表明在用户相似度的计算中, 评分矩阵贡献为 60%, 用户—标签关系的贡献为 40%. 本文所提方法用  $CF_{UI+UT}$  表示, 具体的实验结果如图 3 至 5 所示, 分别为  $a=0.6$  时, 算法的精确度、召回率和  $F$ -measure. 可以看出, 其推荐质量均高于基线方法, 当返回列表选择 Top-15 时, 其  $F$ -measure 的改进达到 6.63%.

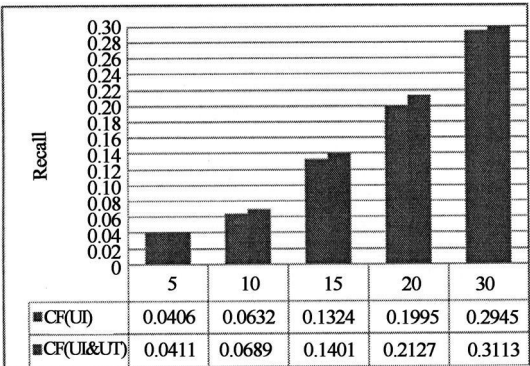


图 3 精确度比较

Fig. 3 Recall comparison

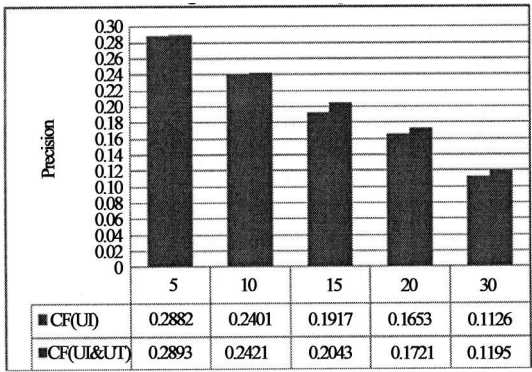


图 4 召回率比较  
Fig. 4 Precision comparison

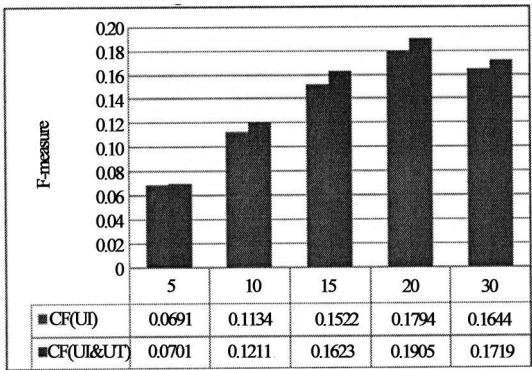


图 5  $F$ -measure 值比较  
Fig. 5  $F$ -measure comparison

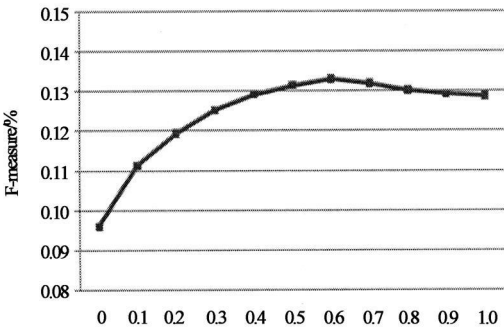


图 6  $F$ -measure 随  $\alpha$  值的变化曲线  
Fig. 6 The change of  $F$ -measure as  $\alpha$

5 总结与展望

本文提供了一个分组同义标签并通过权重相似度混合用户-标签关系到协同过滤的方法. 实验证明了该方法能有效地提高推荐质量. 未来, 我们将进一步探索影响推荐质量的因素, 例如产品和关联组之间的关系、朋友关系和成员关系、用户所处环境等. 另外随着 SNS(Social Network Service) 规模的

爆增, 我们也将关注算法的可扩展性和有效性.

参考文献

[ 1 ] LIU Jian-guo, ZHOU Tao, WANG Bin-hong, *et al.* Progress of the personalized recommendation systems[ J ]. Progress of Nature and Science, 2009, 19(1): 1—15.

[ 2 ] HUANG Zan, CHEN Hsinchun, ZENG Daniel. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering[ J ]. ACM Transactions on Information Systems, 2004, 22(1): 116—142.

[ 3 ] LIU Jian-guo, ZHOU Tao, GUO Qing, *et al.* Overview of the evaluated algorithms for the personal recommendation systems [ J ]. Complex System and Complexity Science, 2009, 6(3): 1—10.

[ 4 ] JIN Rong, SI Luo, ZHAI Cheng-xiang, *et al.* Collaborative filtering with decoupled models for preferences and ratings [ C ]//CIKM '03, November 3—8, New Orleans, Louisiana, USA, 2003.

[ 5 ] RICH E. User modeling via stereotypes[ J ]. Cognitive Science, 1979, 3(4): 329—354.

[ 6 ] GOLDBERG D, NICHOLS D, BRIAN M O, *et al.* Using collaborative filtering to weave an information tapestry [ C ]//Comm ACM, 1992, 35(12): 61—70.

[ 7 ] KONSTAN JA, MILLER BN, DAVID M, *et al.* GroupLens: Applying collaborative filtering to usenet news[ C ]//Comm ACM, 1997, 40(3): 77—87.

[ 8 ] SHARDANAND U, MAES P. Social information filtering: Algorithms for automating ' Word of Mouth' [ C ]//Proc Conf Human Factors in Computing Systems Denver, 1995, 210—217.

[ 9 ] BREESE JS, HECKEMAN D, KADIE C, *et al.* Empirical analysis of predictive algorithms for collaborative filtering[ C ]//Proc 14th Conf Uncertainty in Artificial Intelligence Madison, 1998, 43—52.

[ 10 ] CHEN Yen-liang, CHENG Li-chen. A novel collaborative filtering approach for recommending ranked items[ J ]. Expert Systems with Applications, 2008, 34(4): 2396—2405.

[ 11 ] YANG Ming-hua, GU Zhi-min. Personalized recommendation based on partial similarity of interests[ C ]//Advanced Data Mining and Applications Proceedings, 2009, 509—516.

[ 12 ] ADOM VICIUS G, SANKA RANARAYANAN R, SHAHANA S, *et al.* Incorporating contextual information in recommender systems using a multidimensional approach[ J ]. ACM Trans Inf Syst, 2005, 23(1): 103—145.

[ 13 ] TSO-SUTTER K H L, LEANDRO B M, SHAHANA S, *et al.* Tag-aware recommender systems by fusion of collaborative filtering algorithms[ C ]//SAC' 08 March 16—20, 2008.

[ 14 ] PANAGIOTIS S, ALEXANDROS N, ANNIS M, *et al.* A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis[ J ]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(2): 179—192.

[ 15 ] ZHANG Zi-ke, ZHOU Tao, ZHANG Yi-cheng. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs[ J ]. Physica A: Statistical Mechanics and its Applications, 2010, 22(2): 179—186.

[ 16 ] BENJAMIN M, CIRO C, MENCZER F, *et al.* Evaluating Similarity Measures for Emergent Semantics of Social Tagging [ C ]//WWW 2009, April 20—24, 2009, Madrid, Spain.

[ 17 ] BEGELMAN G, KELLER P, SMADJA F, *et al.* Automated tag clustering: Improving search and exploration in the tag space[ C ]//WWW2006: Proceedings of the Collaborative Web Tagging Workshop, Edinburgh, Scotland, 2006.