一种基于用户偏好自动分类的社会媒体共享和推荐方法

贾大文[1)] 曾 承[1)] 彭智勇[2)] 成 鹏[1)] 阳志敏[1)] 卢 舟[1)]
[1)](武汉大学软件工程国家重点实验室 武汉 430072)
[2)](武汉大学计算机学院 武汉 430072)

摘 要 社会媒体应用已成为Web应用的主流,以用户为中心并且海量媒体数据由用户自生成是社会媒体Web应用的重要特征.应对目前社会媒体环境中信息过载的问题,信息的共享和推荐机制发挥着重要的作用.文中分析了目前主流社会媒体网站基于用户自建组的信息共享机制所存在的问题以及传统推荐技术在效率上的问题,提出了一种新的基于用户偏好自动分类的社会媒体数据共享和推荐方法.直观上讲,该方法的本质是把用户对具体媒体对象的偏好转化成用户对媒体对象所蕴含兴趣元素的偏好,然后把具有相同偏好的用户,即对若干兴趣元素上的兴趣度都相同,自动聚合成为一个"共同偏好组(CPG)".文中提出了基于CPG的社会媒体信息共享和推荐的架构,设计实现了CPG的自动生成算法,通过随机生成模拟数据集实验详细分析了算法性能的影响因素,并与现有类似功能算法进行了效率对比,实验结果表明算法可适用于具有海量用户的社会媒体应用.

关键词 Web数据共享;共同偏好组;社会媒体推荐
中图法分类号 TP311 **DOI号**: 10.3724/SP.J.1016.2012.02381

A User Preference Based Automatic Potential Group Generation Method for Social Media Sharing and Recommendation

JIA Da-Wen[1)] ZENG Cheng[1)] PENG Zhi-Yong[2)] CHENG Peng[1)] YANG Zhi-Min[1)] LU Zhou[1)]
[1)](*State Key of Laboratory of Software Engineering*, *Wuhan University*, *Wuhan* 430072)
[2)](*Computer School*, *Wuhan University*, *Wuhan* 430072)

Abstract Social media applications have become the mainstream of Web application. User-oriented and content generated by users are pivotal characteristics of social media sites. Data sharing and recommendation approaches play an important role in dealing with the problem of information overload in social media environment. In this paper, we analyze the flaws of current group-based information sharing mechanism and the common problem of traditional recommender approaches, and then we propose a novel approach of group automatic generating for social media sharing and recommendation. Intuitively, the essential idea of our approach is that we switch user's preference from the media objects to the interest elements which media objects imply. Then we gather the users who have common preference, namely users have the same interestingness in a set of interest elements, together as Common Preference Group (CPG). We also propose a new social media data sharing and recommendation system architecture based on CPG and design a CPG automatic mining algorithm. By compare our CPG mining algorithm with other algorithm which has similar functionality, it is shown that our algorithm could be applicable to real social media application with massive users.

Keywords Web data sharing; common preference group; social media recommendation

收稿日期:2012-06-05;最终修改稿收到日期:2012-08-20.本课题得到国家自然科学基金(61070011)、湖北省自然科学基金国际合作重点项目、武汉市学科带头人计划项目(201150530139)资助.**贾大文**,男,1982年生,博士研究生,主要研究方向为推荐系统、Web数据管理.E-mail: brilliant@whu.edu.cn. **曾 承**(通信作者),男,1978年生,博士,副教授,主要研究方向为服务计算和社会计算.E-mail: zengc@whu.edu.cn. **彭智勇**,男,1963年生,博士,教授,博士生导师,主要研究领域为复杂数据管理、Web数据管理、可信数据管理等. **成 鹏**,男,1989年生,硕士研究生,主要研究方向为Web服务、云计算. **阳志敏**,男,1989年生,硕士研究生,主要研究方向为Web服务、跨媒体检索. **卢 舟**,男,1990年生,硕士研究生,主要研究方向为Web服务、云计算.

1 引 言

由用户生成内容的社会媒体(Social Media)存在信息过载的问题.社会媒体网站是以用户为中心,具有海量、无序的用户产生的数据.一方面,媒体数据和用户数据都十分庞大,而且不断有新用户加入以及每天都会大量新的数据上传;另一方面,社会媒体数据存在无序的特点,大多数社会媒体是无结构化数据,如视频、图片、文本日志等,这类数据都具有多语义的特点,即每个对象包含多种类别、多粒度的语义信息.社会媒体和网页一样具备海量信息的规模,但又无法直接应用现有成熟的网页信息检索技术对其进行排序,这给 Web 社会媒体数据的分类以及检索带来困难.在线共享和推荐的方式是目前社会媒体信息传播的主要途径.

为了满足信息共享的需要,大多数社会媒体网站提供了用户自建组的机制.文献[1]指出超过一半的 Flickr 用户把他们的图片放入了至少一个组中,这表明了大量用户有参与组的行为和需要.用户建立或加入组有两个目的,一是为了信息共享,其二是为了社交的目的,同一组里面的成员代表他们有着跟组的主题相关的兴趣和爱好[2].一般来说,一个组代表了一个主题,对这个主题感兴趣的用户可以加入组,成为组的成员并可以上传与主题相关媒体数据到组里面.通过对 Flickr 研究表明,向组中加入图片是图片数据传播的主要原因[3-4].无疑,组机制是一种十分有效的社会媒体信息共享和传播的机制.但目前主流的社会媒体网站所支持的用户自建组机制存在如下问题[5-8]:

(1)多同主题的组.每个用户都能随意创建自己想要的组,这使得和同一个主题相似的组的数目很多.例如,Flickr 中有 28109 个关于"cat"的组,而且数目不断增加。面对如此多同主体的组,用户如何选择合适的组加入成为一个难题;同时由于一个主题存在多个组,具有相同兴趣的用户必将分散到不同的组里面,造成信息不能很好地共享的问题.

(2)媒体数据多语义性.一般来说,一个媒体对象会和多个主题相匹配,而根据上条所描述的,一个主题会和多个组相对应,因此,根据现有组共享机制,一个用户若想充分地共享自己的数据就需要把一个媒体数据人工地加入到不同主题下的不同组里面.

(3)用户的无意识兴趣.用户本身是对某些主题感兴趣,但其自身没有意识到已经存在这样的组可以进行数据的共享和交互.即使用户自身愿意参与到组里面进行信息共享的,仍然存在大量这种无意识的用户错过信息共享的机会.

(4)数据噪音.我们把和组里面和组的主题不相关的数据称为数据噪音.媒体对象是人工加入到组里面的,用户加入数据的随意性使我们无法确保对象是和组的主题相关的,事实上,对象和组的主题不一致的现象很常见.

以上提到的问题给社会媒体信息的共享和传播带来了阻碍.为了解决上述问题,本文提出了一种基于用户偏好的组自动生成方法.与用户手动创建的组进行区分,我们把本文提出的基于用户偏好的系统自动生成组称为"共同偏好组(Common Preference Group,CPG)".本文研究基于以下假设:用户对一个社会媒体对象感兴趣是因为用户对媒体对象所蕴含的一些语义元素感兴趣.在这个假设下,我们把用户对许多单一媒体对象的喜好转化为用户对某个语义元素或语义元素的集合的兴趣.本文也把语义元素称为兴趣元素.每个用户对每个语义元素都可以有自身的喜好程度,我们把同一语义元素上的一种喜好程度称为偏好.如果一系列用户对若干语义元素上的每个元素都具有相同的喜好程度,则称这群用户在这些语义元素上具有共同的偏好,把那些有共同偏好的用户聚集成一个组,也就是共同偏好组(CPG).社会媒体应用环境下,CPG 形成过程可简要描述如下:在社会媒体环境下,用户会对某些媒体对象感兴趣,一般来说,系统会提供某些功能让用户来表达他们的喜好,如星级评分,或设置"喜欢"按键.从每个媒体对象都可以抽取出多个语义元素,那么一个用户对媒体对象感兴趣也就是说用户对蕴含在这些媒体对象下的语义元素感兴趣.通过对每个用户喜欢的媒体对象抽取语义并对语义进行统计分析,可以知道用户对哪些语义喜好以及喜好的程度.直观上说,CPG 就是在某些语义元素上,具有相同喜好程度的用户的集合.

无论是从媒体信息共享推荐的角度还是从用户社交的角度来看,CPG 均与传统自创建组机制不同,CPG 有如下优点:(1)一个 CPG 和一组加权的语义相对应,表明这个 CPG 里面的用户在这些语义上具有相同的兴趣度,因此通过 CPG,用户可以发现自己的偏好以及和自己具备相同偏好的其他用户;

(2)CPG 是自动生成的，系统根据对用户行为分析得到用户兴趣，把用户自动加入到相应的 CPG 中；(3)媒体对象也被自动加入到对应的 CPG 中，而无需手动加入.

本文的贡献有如下几个方面：

(1)分析了目前自建组机制在社会媒体共享方面存在的问题.

(2)提出 CPG 概念以及基于 CPG 的数据共享和推荐系统架构.

(3)设计 CPG 自动生成算法，并提出了基于 CPG 协同兴趣发现的思想.

(4)实现了 CPG 的自动生成算法，通过随机生成模拟数据集实验详细分析了算法性能的影响因素，并和已有类似的算法进行了性能对比. 实验结果表明了该算法可适用于具有海量用户的社会媒体应用.

本文第 2 节给出 CPG 相关问题和定义的描述；第 3 节提出基于 CPG 的社会媒体数据共享和推荐的系统架构；第 4 节阐明 CPG 的生成过程，设计并实现一种 CPG 自动生成算法；第 5 节通过大量的模拟实验来分析各种参数对算法复杂度的影响，同时还将与类似功能算法做性能比较；第 6 节对本文相关工作进行讨论；第 7 节总结全文并给出后继工作的想法.

2　概念定义

本节将给出相关的主要概念和形式化的定义. 对于传统推荐系统包括的主要概念有用户(*User*)、对象(*Object*)、评分(*Rating*)、归档信息(*Profile*)和用户-对象矩阵(User-Object Matrix，***UOM***). 整个推荐过程是通过用户对初始对象的评分建立用户的 *Profile* 来预测用户对其它对象的评分. 系统根据预测结果对待推荐对象进行排序并确定是否把对象推荐给该用户. 构建 ***UOM*** 的过程是推荐系统的基础.

定义 1.　用户-对象矩阵(***UOM***). 用户对象矩阵记录用户对对象的评分，表示用户对象的喜好程度. ***UOM*** 可以表示成三元组〈*User*，*Object*，*Rating*〉，其中，*User* 表示用户的集合；*Object* 表示对象的集合；*Rating* 表示用户对对象的评分的所有取值. ***UOM*** 可以看成一个从用户和对象到评分值的函数.

$$\boldsymbol{UOM}: User \times Object \rightarrow Rating.$$

我们前面提到了目前社会媒体具有海量用户和数据的特点，并且用户和数据的数量每天快速增长. 本文提出了基于共同偏好组(Common Preference Group，CPG)的信息共享和推荐方式，本文新增加了几个概念：兴趣元素(*Interest-Element*)、兴趣度(*Interestingness*)和用户-兴趣元素矩阵(User-Element Matrix，***UEM***)，共同偏好组(Common Preference Group，CPG)，组偏好关系-对象矩阵(PGR-Object Matrix，***POM***). 通过对 ***UEM*** 挖掘生成 CPG，然后构建 ***POM*** 进行推荐.

定义 2.　兴趣元素(*Interest-Element*). 兴趣元素是包含在对象里面的能够影响到用户对该对象兴趣的特征或语义元素. 不同兴趣元素之间可以有组合关系和泛化关系.

定义 3.　兴趣度(*Interestingness*). 兴趣度是描述用户对每个兴趣元素感兴趣的程度的量化值.

定义 4.　用户-兴趣元素矩阵(***UEM***). 用户兴趣元素矩阵记录每个用户对每个兴趣元素的兴趣度. ***UEM*** 可以表示成三元组〈*User*，*Interest-Element*，*Interestingness*〉，其中，*User* 表示用户的集合；*Interest-Element* 表示兴趣元素的集合；*Interestingness* 表示用户兴趣度的有所取值. ***UEM*** 可以看成一个从用户和兴趣元素到兴趣度的函数.

$$\boldsymbol{UEM}: User \times Interest\text{-}Element \rightarrow Interestingness.$$

定义 5.　共同偏好组(*CPG*). 一个共同偏好组是 ***UEM*** 中的子集，这个子集满足以下条件，同一个 *CPG* 里面的所有用户对这个 *CPG* 里面的每个兴趣元素都有相同的兴趣度. 生成的 *CPG* 包括两个关系，组-用户关系(User-Group Relation，*UGR*)表示 *CPG* 所包含的用户，组偏好关系(Preference-Group Relation，*PGR*)表示 *CPG* 所包含的兴趣元素及其兴趣度.

定义 6.　共同偏好组-对象矩阵(***GOM***). 共同偏好组-对象矩阵记录共同偏好组和对象之间的相似性. ***GOM*** 可以表示成三元组〈*CPG*，*Object*，*Similarity*〉，其中，*CPG* 表示共同偏好组的集合；*Object* 表示对象的集合；*Similarity* 表示 *CPG* 的组偏好关系和目标对象特征的相似性. ***GOM*** 可以看成一个从 CPG 和对象到相似度的函数.

$$\boldsymbol{GOM}: CPG \times Object \rightarrow Similarity.$$

显然，共同偏好组的数量将远远少于用户数，即和传统用户-对象矩阵相比较，***GOM*** 的计算复杂度将会大大降低. 这种推荐模式将适用于海量用户和数据的环境.

3 基于 CPG 的社会媒体共享和推荐系统架构

为了弥补传统基于用户自建组的信息共享方式和传统推荐系统的问题，本节提出了一种基于用户偏好组的社会媒体数据共享和推荐的架构，如图 1 所示.

我们可以想象每一个媒体对象基本上都是一个独一无二的个体，然而其所包含的让用户感兴趣的兴趣元素却具有共同性. 本研究基于的假设是用户对一个社会媒体对象感兴趣是因为用户对媒体对象所蕴含的兴趣元素感兴趣. 在目前社会化网络环境下，我们通过分析用户行为，很容易得到用户与对象之间的喜好关系. 通过前面所提到的假设，我们的目标是把用户和对象之间的喜好关系转化为更具有一般性的用户和兴趣元素之间的喜好关系，建立用户兴趣元素矩阵（***UEM***）. 通过 ***UEM***，可自动生成 CPG. 同一个 CPG 里面的所有用户符合一个共同的偏好，即对系列兴趣元素都具备同样的兴趣度. 用户通过社会化网络上的行为被自动分类到了一起，而且知道之所以被聚到一起的原因，即具备什么样的共同偏好. 待推荐对象包括媒体对象和新加入系统的用户. 先通过提取媒体对象或新用户的兴趣特征，推荐算法可以通过待推荐兴趣特征，同 CPG 中的 *PGR* 做相似性比较，最后确定是否把待推荐信息加入到相应的 CPG 中.

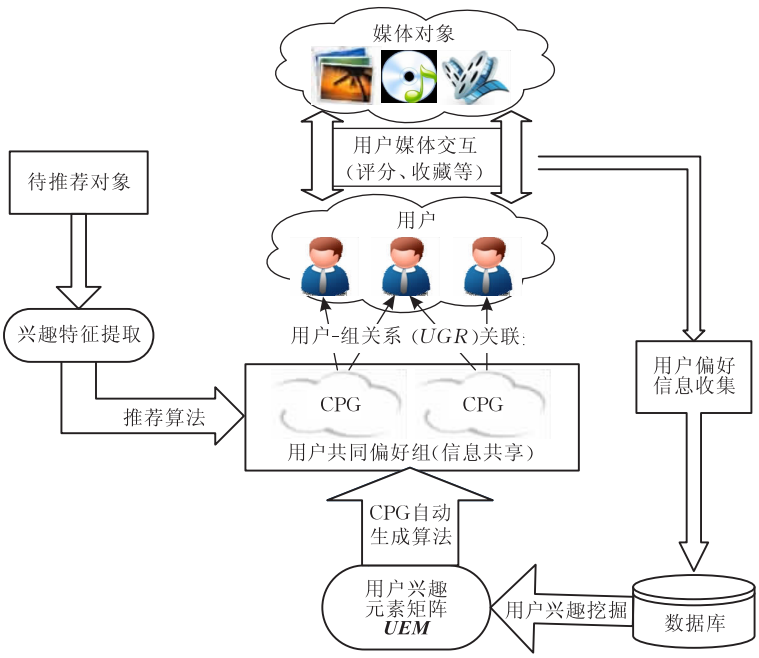


图 1 基于 CPG 的信息共享和推荐系统架构

4 CPG 自动生成

CPG 自动生成的过程即从 ***UEM*** 找出对应 *UGR* 和 *PGR* 的过程. 我们定义 *minUsers* 来表示构成一个 CPG 所要求的最少用户的数目.

4.1 建立用户兴趣元素矩阵

用户兴趣元素矩阵（***UEM***）记录每个用户对每个兴趣元素的兴趣度. 从媒体数据对象中抽取蕴含用户偏好的兴趣元素来构建 ***UEM*** 是此方法的基础. 兴趣元素的抽取过程需要考虑两个方面：（1）兴趣元素可以通过预定义或通过抽取媒体对象语义的方式获得. 不管采取哪种方式，兴趣元素应该与用户真实兴趣相符合；（2）媒体对象兴趣元素抽取的过程要尽可能地准确. *IESpace* 表示所有让用户感兴趣的兴趣元素的集合，o_e 表示对象 o 所蕴含的兴趣元素.

$$o_e \subseteq IESpace \tag{1}$$

一个用户会和若干媒体对象交互. *ObjectSpace* 表示所有的媒体对象，u_{favorite} 表示用户喜好的对象集合.

$$u_{\text{favorite}} \subseteq ObjectSpace \tag{2}$$

基于式（1）、（2），用户偏好 $u_{\text{preference}}$ 可以推导出来，一个用户 u 的偏好 $u_{\text{preference}}$ 是指用户 u 所感兴趣的所有兴趣元素以及对每个兴趣元素兴趣度 λ_i.

$$u_{\text{preference}} \subseteq \{\langle e_i, \lambda_i \rangle\} \tag{3}$$

根据式(3)，我们可以建立用户偏好矩阵.

4.1.1　兴趣度的确定方法

为了区分不同用户对同一兴趣元素喜好的差异性，本文提出了兴趣度的概念. 兴趣度，即用户对兴趣的偏爱喜好程度. 完成用户的兴趣发现之后，针对不同的兴趣，结合每个用户与所有推荐信息的交互次数来衡量用户在该兴趣上的兴趣程度. 兴趣程度与兴趣度的关系是兴趣程度是一个绝对的数值，而兴趣度是一个相对的兴趣等级. 用户 u_i 在兴趣元素 e 上的兴趣程度 θ_i^e 的计算公式为

$$\theta_i^e=\frac{IN_i^e}{IF_i+\alpha} \tag{4}$$

其中，IN_i^e 表示用户 u_i 与兴趣元素 e 的交互次数；IF_i 表示用户 u_i 的交互频率；α 用来避免出现除零情况.

$$IF_i=\frac{sumIN_i}{avgIN} \tag{5}$$

其中，$sumIN_i$ 表示用户 u_i 的总交互次数；$avgIN$ 表示平均每个用户的交互次数. 交互频率是为了平衡用户行为习惯对真实兴趣程度的影响. 兴趣度 λ 由量化式(4)中的用户对某个兴趣元素的兴趣程度 θ 值得到.

4.2　CPG 生成算法

本小结给出一个简单的例子直观地说明 CPG 生成的原理(如图 2 所示)，并详细介绍算法的过程. 本例中定义用户集 $U=\{u_1,u_2,u_3,u_4,u_5\}$，兴趣元素集 $E=\{e_1,e_2,e_3,e_4,e_5\}$，用户对兴趣元素的兴趣度量化成为 3 种 $\{\lambda_X,\lambda_Y,\lambda_Z\}$，本例给定 $minUsers=2$. 如图 2(a)所示，用户对每个兴趣元素的兴趣度都记录在 ***UEM*** 中. 下一步是挖掘 CPG 集合，如图 2(b)所示，不同形状的区域代表不同的 CPG，本例中有 3 个CPG. 根据 CPG，组-用户关系和组偏好关系将很容易推导出来，分别如图 2(c)和图 2(d)所示.

	e_1	e_2	e_3	e_4	e_5
u_1	λ_X	λ_Y	λ_X		
u_2				λ_Y	λ_Y
u_3	λ_X	λ_Y	λ_X	λ_Y	λ_Y
u_4	λ_Z	λ_Z			
u_5	λ_Z	λ_Z		λ_Y	λ_Y

(a) 用户兴趣元素矩阵(***UEM***)

	e_1	e_2	e_3	e_4	e_5
u_1	λ_X	λ_Y	λ_X		
u_2				λ_Y	λ_Y
u_3	λ_X	λ_Y	λ_X	λ_Y	λ_Y
u_4	λ_Z	λ_Z			
u_5	λ_Z	λ_Z		λ_Y	λ_Y

(b) 共同兴趣组(*CPG*)

	e_1	e_2	e_3	e_4	e_5
CPG_1	λ_X	λ_Y	λ_X		
CPG_2				λ_Y	λ_Y
CPG_3	λ_Z	λ_Z			

(c) 组偏好关系(*PGR*)

	u_1	u_2	u_3	u_4	u_5
CPG_1	1		1		
CPG_2		1	1		1
CPG_3				1	1

(d) 组用户关系(*UGR*)

图 2　一个 CPG 挖掘的例子

本节给出了具体 CPG 自动生成算法，并基于图 2 详细描述算法过程.

第 1 步. 把用户按照其偏好聚集，即把每种兴趣元素下，不同兴趣度的用户都聚成一个单元.

通过用户兴趣元素矩阵 ***UEM***，很容易可以得到用户聚集表(如表 1 所示). 对于表中每个字段 p，如果字段 p 包含的用户数大于或等于 *minUsers*，则把 p 加入 *initialList* 中，否则就从表中删除. 最后 *intialList* 中的字段可以看成满足条件的只包含单个兴趣元素的 CPG.

表 1　用户聚集表

	e_1	e_2	e_3	e_4	e_5
λ_X	u_1,u_3		u_1,u_3		
λ_Y		u_1,u_3		u_2,u_3,u_5	u_2,u_3,u_5
λ_Z	u_4,u_5	u_4,u_5			

算法 1.　UserClustering.

输入：***UEM***(用户兴趣元素矩阵)

输出：*initialList*

描述：把 ***UEM*** 转化成用户聚集表，表里面每个字段 p 都表示在某一兴趣元素 e_i 上有相同兴趣度 λ_j 的用户的集合.

1. 从 ***UEM*** 中抽取用户集合 U，兴趣元素集合 E 和兴趣度集合 Λ；
2. for 每个兴趣元素 $e_i\in E$
3.　for 每个兴趣度 $\lambda_j\in\Lambda$
4.　if (用户 u 对 e_i 的兴趣度为 λ_j)
5.　把用户 u 加入字段 $p(e_i\lambda_j)$ 中；
6.　end if
7.　end for
8.　if ($p(e_i\lambda_j)$ 中用户数目大于等于 *minUsers*)
9.　把 $p(e_i\lambda_j)$ 加入到 *initialList* 中；
10.　end if
11. end for

第 2 步. 生成包含多个兴趣元素的 CPG.

对 *initialList* 中的每个字段 $p(e_i\lambda_j)$，我们将其与表 1 中位于其所在列后面的列的有效字段(即字段中所包含的用户数大于或等于 *minUsers*)进行 $\cap$ 操作. $\cap$ 操作是指两个字段 p 中的偏好 $e_i\lambda_j$ 的并集以及两个字段所包含用户 u 的交集.

以第一个字段 $p(e_1\lambda_X)$ 为例，表 2 给出了 $p(e_1\lambda_X)$ 与表 1 中位于 $p(e_1\lambda_X)$ 所在列后面列中有效字段进行 $\cap$ 操作后得到的用户交集结果. 删除结果中用户数目小于 *minUsers* 的字段，把结果加入

$intersectionList(e_i\lambda_j)$中.定义 $CCL(e_i\lambda_j)$(Candiate CPG List)来存放对 *initialList* 中的每个字段 $p(e_i\lambda_j)$处理后所生成的候选多兴趣元素 CPG.把 $intersectionList(e_i\lambda_j)$中的每个元素与 $CCL(e_i\lambda_j)$中的所有元素依次进行∩操作后的结果加入$CCL(e_i\lambda_j)$中,每次循环的最后 $intersectionList(e_i\lambda_j)$中的元素自身也需要加入到 $CCL(e_i\lambda_j)$中.

表 2 字段 $p(e_1\lambda_X)$与其它字段∩操作的结果

	e_1	e_2	e_3	e_4	e_5
λ_X	u_1,u_3		u_1,u_3		
λ_Y		u_1,u_3		$u_{3(\text{delete})}$	$u_{3(\text{delete})}$
λ_Z					

算法 2. CreateCandidateCPGList.

输入:p, *initialList*

输出:*CandiateCPGList*

描述:对 *initialList* 中的每个字段 p,将其与表 1 中其它列的元素作∩操作,生成候选 CPG.

1. $CCL(p)=\{\}$;
2. for(每个 *initialList* 中的字段 p_j)并且(p_j的兴趣元素在用户聚集表的顺序在输入字段 p 的兴趣元素的后面)
3. $users=p_j.users\cap p.users$;
4. $preferences=p_j.preferences\cup p.preferences$;
5. if(*users* 数目大于 *minUsers*)
6. 将元素$\langle\{preferences\},\{users\}\rangle$加入到 $intersectionList(p)$;
7. end if
8. for $intersectionList(p)$的每个元素 t_i
9. $MergeCandidateCPGList(t_i,CCL(p))$;
10. 将 t_i加入 $CCL(p)$;
11. $RemoveOverlapped(CCL(p))$;
12. end for
13. Return $CCL(p)$;
14. end for

以字段 $p(e_1\lambda_X)$为例,$intersectionList(e_1\lambda_X)=\{\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle,\langle\{e_1\lambda_X,e_3\lambda_X\},\{u_1,u_3\}\rangle\}$.

(1)对元素$\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle$,先将其与 $CCL(e_1\lambda_X)$合并,但目前 $CCL(e_1\lambda_X)$为空,无法生成新的多兴趣元素 CPG,所以仅把元素$\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle$本身加入到 $CCL(e_1\lambda_X)$.此时$CCL(e_1\lambda_X)=\{\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle\}$.

(2)对元素$\langle\{e_1\lambda_X,e_3\lambda_X\},\{u_1,u_3\}\rangle$,先与目前 $CCL(e_1\lambda_X)$中的元素$\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle$进行∩操作.前面定义了∩操作是指对元素的偏好集合求并集,用户集合求交集,其目的是检测当前元素能否为目前的候选 CPG 带来更多的兴趣元素(算法 2 中,*MergeCandidateCPGList* 用于实现这一功能).结果产生了新元素$\langle\{e_1\lambda_X,e_2\lambda_Y,e_3\lambda_X\},\{u_1,u_3\}\rangle$,将其加入 $CCL(e_1\lambda_X)$,同时还要将元素$\langle\{e_1\lambda_X,e_3\lambda_X\},\{u_1,u_3\}\rangle$本身也加入 $CCL(e_1\lambda_X)$.$CCL(e_1\lambda_X)=\{\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle,\langle\{e_1\lambda_X,e_2\lambda_Y,e_3\lambda_X\},\{u_1,u_3\}\rangle,\langle\{e_1\lambda_X,e_3\lambda_X\},\{u_1,u_3\}\rangle\}$.此时我们发现元素$\langle\{e_1\lambda_X,e_2\lambda_Y,e_3\lambda_X\},\{u_1,u_3\}\rangle$完全覆盖了元素$\langle\{e_1\lambda_X,e_2\lambda_Y\},\{u_1,u_3\}\rangle$和$\langle\{e_1\lambda_X,e_3\lambda_X\},\{u_1,u_3\}\rangle$,元素 A 被元素 B 覆盖是指,$A.preference\subseteq B.preference$ 并且$A.users\subseteq B.users$.所以如果 *CCL* 中的一个元素被另一个元素所覆盖,需要从中删除被覆盖元素(算法 2 中,*RemoveOverlapped* 用于实现这一功能).

第 3 步.对 *initialList* 中每个元素,重复步 2.

对 *initialList* 中的每个元素,我们都通过重复步 2 得到一个对应的 *CCL*,如表 3 所示.每个 *CCL* 中的候选 CPG 已经消除了覆盖的情况,但由不同字段产生的候选 CPG 还是可能存在覆盖,所以需要查找出被覆盖的元素并删除(*RemoveRedundent* 用于消除不同 *CCL* 间元素的覆盖).

表 3 所有候选 CPG 列表

P	*CCL*
$p(e_1\lambda_X)$	$\langle\{e_1\lambda_X,e_2\lambda_Y,e_3\lambda_X\},\{u_1,u_3\}\rangle$
$p(e_1\lambda_Z)$	$\langle\{e_1\lambda_Z,e_2\lambda_Z\},\{u_4,u_5\}\rangle$
$p(e_2\lambda_Y)$	$\langle\{e_2\lambda_Y,e_3\lambda_X\},\{u_1,u_3\}\rangle(Delete)$
$p(e_2\lambda_Z)$	$\varnothing$
$p(e_3\lambda_X)$	$\varnothing$
$p(e_4\lambda_Y)$	$\langle\{e_4\lambda_Y,e_5\lambda_Y\},\{u_2,u_3,u_5\}\rangle$
$p(e_5\lambda_Y)$	$\varnothing$

我们可以看到由字段$p(e_2\lambda_Y)$产生的 $CCL(e_2\lambda_Y)$被已有的$\langle\{e_1\lambda_X,e_2\lambda_Y,e_3\lambda_X\},\{u_1,u_3\}\rangle$覆盖,所以需要删除.实际上,如果字段 p'中的用户是已经计算过的字段 p 的子集,那么由字段 p'生成的 *CCL* 必然被字段 p 所生成的 *CCL* 覆盖.最后,我们把所有 *CCL* 产生的无覆盖元素加入 *CPGList* 中.有了 *CPGList*,可以很容易得到 *UGR* 和 *PGR*.

算法 3. CreateCPGList.

输入:***UEM***(用户兴趣元素矩阵)

输出:*CPGList*

1. $CPGList=\{\}$;
2. $initialList=UserClustering(\boldsymbol{UEM})$;
3. for each element $p\in initialList$
4. $CCL(p)=CreateCandidateCPGList(p)$;
5. 将 *CCL* 加入到 *tempList*;

```
6. end for
7. for each CCL(p_i)∈tempList
8.    for each CCL(p_j)∈tempList
9.       if p_j.e after p_i.e
10.         RemoveRedundent(CCL(p_i),CCL(p_j));
11.      end if
12.   endfor
13. endfor
14. CPGList=tempList;
15. Return CPGList.
```

4.3　基于协同的 CPG 合并思想

前面的一节详细讨论了 CPG 自动生成的具体过程.算法从 ***UEM*** 中挖掘出所有的 CPG.这就可能出现一种情况就是,很多的 CPG 只有少量用户和用户偏好的差异,如图 3 所示,CPG_1 和 CPG_2 仅在兴趣元素 e_1 上存在少量差异.如果大量这类 CPG 存在的话会造成 CPG 数目变得很大,由第 2 节给出的定义 6 可知,生成 CPG 的目的之一是改变传统通过直接比较用户对对象的评分的关系,从而提高推荐效率,而如果类似这种大同小异的 CPG 数目很多的话,效率就得不到保证.所以这里采用协同过滤思想对 CPG 进行合并,CPG 合并不仅能减少 CPG 的数目以提高推荐效率,同时还能发现用户潜在可能的兴趣偏好,以提高推荐的新颖度.同时,用户能通过这种途径发现潜在的新的兴趣喜好.

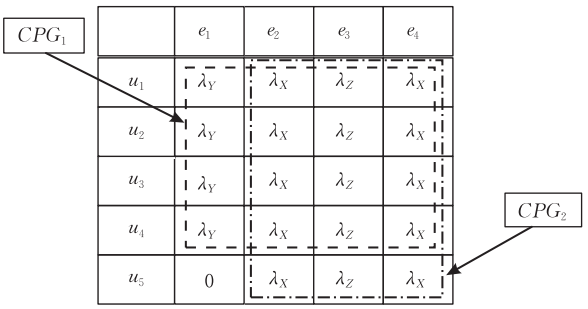


图 3　CPG 合并情形

当两个 CPG 的相似度值 $H(CPG_1,CPG_2)$ 高于一定阈值 ψ,两个 CPG 将合并成一个 CPG,一种简单的合并方法就是找出同时包含两 CPG 最小矩形,这就意味着新的 CPG 中的某些用户以前不存在的兴趣元素会得到自动填充.

5　实验结果与分析

我们用 Java 实现了 CPG 自动生成算法,并设计了一系列的实验来对算法效率进行了分析.实验环境为 2.8GHz CPU 和 4GB 内存的 PC,Windows 7 操作系统.

5.1　CPG 生成算法性能分析

本小结对提出的 CPG 生成算法的效率进行详细的分析.采用生成随机数据的方法构建模拟数据集,分别生成稠密用户兴趣元素矩阵 $\boldsymbol{M}_d$ 和稀疏用户兴趣元素矩阵 $\boldsymbol{M}_s$. $\boldsymbol{M}_d$ 中的每个元素均随机填充兴趣度值,而 $\boldsymbol{M}_s$ 中为每个用户随机填充 5～10 个兴趣元素的兴趣度,其它为空.

根据 CPG 生成算法的原理,有 4 个参数〈U,E,N,$minUsers$〉会对算法性能产生影响,其中,U 指用户数目;E 是兴趣元素数目;N 是指不同兴趣度的个数;$minUsers$ 指生成 CPG 所需要包含的最少用户数目.

本文分别分析 CPG 自动生成算法对这 4 个参数效率方面的敏感度.如图 4 所示,图中纵坐标代表毫秒(ms)级别的时间开销.

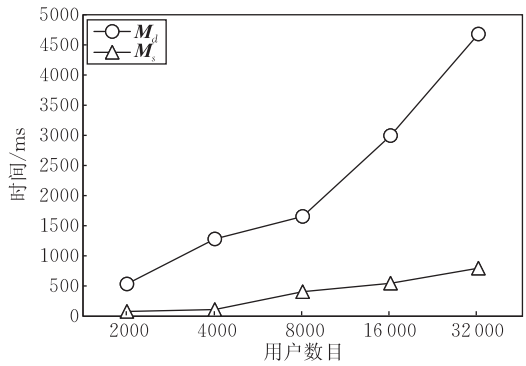


图 4　用户数目对算法性能的影响
($minUsers=50$,$E=50$,$N=3$)

图 4 表示算法效率与用户数量 U 的关系.可以看出,当用户数目成倍迅速增长时,时间开销增长仍呈较慢的线性增长,这说明算法对用户数目的容忍度高,特别是在计算矩阵中数据稀疏的条件下.实际应用中,数据将会是稀疏的,说明算法将有能力处理实际中大规模用户条件下 CPG 的自动生成的任务.

图 5 表示兴趣元素的数目 E 对算法性能的影响.其它参数条件相同的情况下,左侧主坐标轴表示稠密矩阵 $\boldsymbol{M}_d$ 的时间开销,而右侧次坐标轴表示稀疏矩阵 $\boldsymbol{M}_s$ 的时间开销.针对 $\boldsymbol{M}_d$,兴趣元素数目 E 的增加意味着整个矩阵中增加了更多的元素,每增加一列元素都可能导致与其它列中的元素组合出新

的 CPG，换言之，兴趣元素数目 E 的变化对算法开销影响很大. 然而，对于 $\boldsymbol{M}_s$，每个用户仅随机填充 5～10 个兴趣元素的兴趣度，即整个矩阵中元素个数相对固定，E 增加只会使得矩阵变得更加稀疏. 当矩阵稀疏度达到一定程度后，能组合为满足条件的 CPG 的数目将会变少. 正如图中所示，随着 E 的增长，算法的实际开销反而会降低. $\boldsymbol{M}_d$ 与用户对所有兴趣元素都感兴趣的极端情况相对应，由于用户在一定时期内的兴趣是稳定的，真实数据的分布会更加接近 $\boldsymbol{M}_s$. 不同的是，用户的兴趣数目可能不会均匀分布.

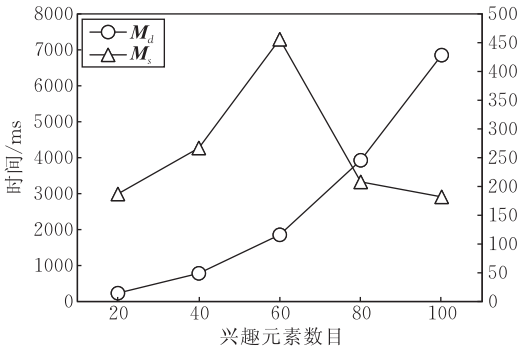


图 5 兴趣元素数目对算法性能的影响
($minUsers=20, U=5000, N=33$)

图 6 表示参数 *minUsers* 不同情况下的算法时间复杂度. 随着设定的 CPG 最少用户量的提高，算法第 1 步中 *intialList* 过滤掉的元素以及第 2 步中 *intersectionList* 过滤掉的元素则越多，算法速度将会越快. 当最少用户数大到一定程度，算法复杂度趋于稳定.

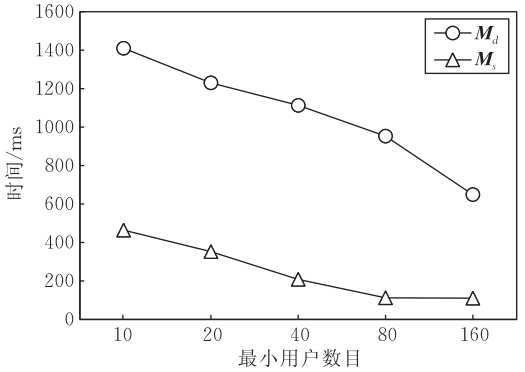


图 6 最小用户数目对算法性能的影响
($U=5000, E=50, N=3$)

图 7 表示不同兴趣度的个数对性能的影响. 算法开销随着兴趣度个数的增加呈现出先递增后再递减的现象. 而峰值出现的原因是当兴趣度区分不明显时，多数用户会被划分到同一 CPG 中，导致算法效率提高；但是，当兴趣度个数超过峰值后，由于差异性过大，会导致满足条件的 CPG 大幅减少，从而使得算法时间开销急剧下降.

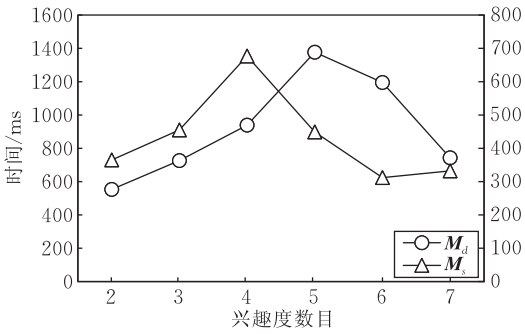


图 7 兴趣度数目对算法性能的影响
($\boldsymbol{M}_d$：$minUsers=100, U=5000, E=50$，
$\boldsymbol{M}_s$：$minUsers=30, U=10000, E=50$)

下面本文用一组实验来分析兴趣度峰值大小与参数之间的关系. 对 $\boldsymbol{M}_d$ 而言，兴趣度峰值呈现如下规律：

(1) 峰值与最小用户数目的关系

用户数为 5000，兴趣元素为 50

最小用户数	兴趣度峰值
50	8
100	5
150	4

结果说明兴趣度峰值与最少用户数成反比.

(2) 峰值与兴趣元素数目的关系

用户数为 5000，最少用户为 100

兴趣元素	兴趣度峰值
20	5
50	5
80	5

结果说明兴趣度峰值与兴趣元素个数无关.

(3) 峰值与用户数目的关系

最少用户为 100，兴趣元素为 50

用户数目	兴趣度峰值
2000	3
5000	5
10000	8

结果说明兴趣度峰值与用户数目成正比.

对 $\boldsymbol{M}_s$ 矩阵而言，兴趣元素个数也会对峰值产生影响. 如前文所述，兴趣元素个数决定了 $\boldsymbol{M}_s$ 矩阵的稀疏度. 当稀疏度一定时，上文结论仍然适用.

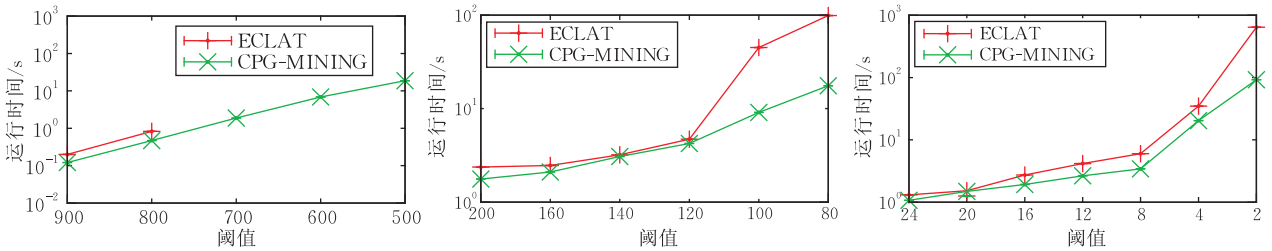
最终实验得到的结果是，兴趣度峰值的大小与用户数目和最小用户数目的比值相关，并且也与 ***UEM*** 的稀疏度相关.

5.2　相关算法对比分析

本文提出的 CPG 自动生成算法和关联规则问题中频繁闭项集的挖掘(Frequent Closed Itemset Mining)算法有类似之处[9]. 不同之处在于,那些算法的输入相当于一个布尔类型的矩阵,而本文算法输入的用户兴趣元素矩阵 ***UEM*** 中带有不同的兴趣度值,布尔矩阵可以看成是兴趣度个数只有一个的特例. 传统关联规则问题中的支持度 *Support* 的语境可以和本文提出的 CPG 自动生成问题中最小用户数 *minUsers* 与用户总数 U 的比值对应. 在本文应用中,关联规则问题中的置信度 *Confidence* 不需要考虑.

本文将 CPG 自动生成算法与其类似算法 ECLAT[11] 的频繁闭项集实现[10] 进行了效率对比实验. 由稠密到稀疏,我们分别选取了 3 种数据集来进行算法性能对比. 分别是稠密真实数据集 Chess 以及两组稀疏数据集 T40I10D100K,T10I4D100K. 实验数据为从 Chess 数据集中选取前 1000 条记录,从 T40I10D100K 数据集中选取前 10 000 条记录以及 T10I4D100K 选取了前 20 000 条记录. 选取的数据集可以直接在 ECLAT 实现算法上运行,同时需要转换为我们算法中的 ***UEM*** 结构再使用 CPG-MINING 算法运行. 阈值对 ECLAT 算法而言是表示生成频繁项所需要的最少记录数目,决定了支持度的大小,对我们的算法(CPG-MINING)而言是指最小用户数.

图 8 给出了这两种算法的性能对比结果,从图中可以看出,ECLAT 算法不适合稠密数据集,且对支持度的改变比较敏感,如图 8(a)所示,对于较为稠密的 Chess 数据集,支持设定低于 0.8(即图(a)中阈值小于 800)时,算法将无法得到结果;对较为稀疏的数据集,如图 8(b)、(c)所示,随着支持度的降低,ECLAT 算法运行时间将急剧增长. 结果表明我们提出的算法在各种条件下的效率均优于类似算法 ECLAT 的效率.



(a) 数据集Chess　(b) 数据集T40I10D10K　(c) 数据集T10I4D100K

图 8　实验结果对比

6　相关工作

就我们调研的情况来看,目前还没有关于自动生成用户兴趣组的研究. 本文与两个方面的研究进行比较来体现本文的意义.

6.1　组推荐技术

因为社会媒体的组是用户自定义生成的,所以一个主题对应多个组,如前面所提到的,组的数目甚至可以达到上万个. 因此,对一个用户而言,如果其对一个主题感兴趣,那么他不可能加入所有相关的组里面去,对于用户而言,很难确定把媒体对象发布到哪个组里面以及如何找到适合自己的组. 许多研究关注于对象与组的匹配[6-8],还有一些研究关注把组推荐给用户[4,12]. 这些推荐机制优点是找出最适合的 Top-k 的组推荐给用户或媒体对象,但其仍然不能满足媒体对象充分共享的任务,具有相同兴趣的用户仍然分离在不同的组里面,组与组之间对象不能共享. 另一方面,某些用户可能明显具备相同的兴趣,但他们忽视了这样的组的存在,结果他们不能认识彼此,信息也不能共享. 基于 CPG 的方法,媒体和用户是自动分配到相应的 CPG 之中.

6.2　社会媒体推荐技术

从社会媒体推荐技术的角度来考虑,研究着重于推荐算法的准确性方面. 这些推荐方式包括基于内容的(Content-based)[13-14]、协同过滤的(Collaborating Filtering)[15-17]、人口统计学的(Demographic)[18] 以及混合式的(Hybrid)[19-20] 等推荐方法. 文献[21]提出了基于内容/相似度的推荐技术用于社会媒体推荐. 当将现有技术用于社会媒体推荐,我们发现了一个普遍的问题,现有技术都是把单个对象推荐给单个用户,但社会媒体具有海量用户以及海量数据的特点,并且用户和媒体的数目每天都在不断增加,这些方法主要考虑推荐算法的准确性而忽略了推荐的效率问题. 尽管时间效率问题在推荐系统里面不像在检索系统里面那么关键,但我们认为在海量数据

环境下考虑推荐算法的时间效率是十分必要的.显然,共同偏好组的数量将远远少于用户数,即和传统用户-对象矩阵相比较,共同偏好组-对象矩阵的计算复杂度将会大大降低,这种推荐模式将适用于海量用户和数据的环境.

7 总结与展望

本文分析了目前社会媒体网站中组机制的不足以及相关的推荐技术,提出了一种基于用户偏好的潜在组自动生成机制用于社会媒体数据的共享和推荐,设计并实现了 CPG 自动生成方法,并通过实验详细分析影响算法性能的各种因素.实验结果表明该算法的正确性与高效性,可适用于具有海量用户的社会媒体应用.同时,本文提出了基于 CPG 的数据共享和推荐系统架构,如要真正基于此架构实现具备实用性的推荐系统还需继续研究完善,解决如下若干问题:(1)实现从用户行为得到实际的用户偏好矩阵的过程,包括兴趣元素的确立、社会媒体语义挖掘以及兴趣度的确立;(2)推荐算法的研究,包括用户的自动加入和退出机制、待推荐媒体数据和 CPG 的推荐匹配算法以及 CPG 的演变机制;(3)进一步提高 CPG 生成算法的效率以适应海量用户和提高更细粒度兴趣元素条件下的算法效率.此外,我们将进一步思考 CPG 用于信息共享和推荐的其它适用环境.

参 考 文 献

[1] Negoescu R A, Gatica-Perez D. Analyzing Flickr groups//Proceedings of the 7th ACM International Conference on Image and Video Retrieval. Baoding, China, 2008: 417-426

[2] Yu Jie, Jin Xin, Han Jia-Wei, Luo Jie-Bo. Collection-based sparse label propagation and its application on social group suggestion from photos. ACM Transactions on Intelligent Systems and Technology, 2011, 2(2): 12:1-12:21

[3] Lerman K, Jones L. Social browsing on Flickr//Proceedings of the International Conference on Weblogs and Social Media. Boulder, Colorado, USA, 2007: 210-230

[4] Zwol R V. Flickr: Who is looking?//Proceedings of the ACM International Conference on Web Intelligence. Silicon Valley, California, USA, 2007. New York: ACM, 2008: 184-190

[5] Zheng Nan, Li Qiu-Dan, Liao Sheng-Cai, Zhang Lei-Ming. Flickr group recommendation based on tensor decomposition//Proceedings of the 33nd ACM Special Interest Group on Information Retrieval. Singapore, 2008: 737-738

[6] Negoescu R A, Gatica-Perez D. Topickr: Flickr groups and users reloaded//Proceedings of the 16th ACM International Conference on Multimedia. Vancouver, Canada, 2008: 857-860

[7] Cai Jun-Jie, Zha Zheng-Jun, Qi Tian, Wang Zeng-Fu. Semi-automatic Flickr group suggestion//Proceedings of the 17th International Multimedia Modeling Conference. Taibei, China, 2011: 77-87

[8] Chen Hong-Ming, Chang Ming-Hsiu, Chang Ping-Chieh, Tien Ming-Chun, Winston H Hsu, Wu Ja-Ling. SheepDog: Group and tag recommendation for Flickr photos by automatic search-based learning//Proceedings of the 16th ACM International Conference on Multimedia. Vancouver, British Columbia, Canada, 2008: 737-740

[9] Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed itemsets for association rules//Proceedings of the 7th International Conference on Database Theory. Jerusalem, Israel, 1999: 398-416

[10] Borgelt C. Efficient implementations of Apriori and Eclat//Proceedings of the IEEE International Conference on Data Mining Workshop Frequent Itemset Mining Implementations. Melbourne, USA, 2003

[11] Zaki M J, Parthasarathy S, Ogihara M, Li Wei. New algorithms for fast discovery of association rules//Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. Newport Beach, USA, 1997: 283-296

[12] Zheng Nan, Li Qiu-Dan, Liao Sheng-Cai, Zhang Lei-Ming. Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr. Journal of Information Science, 2010, 36(6): 733-750

[13] Bu Jia-Jun, Tan Shu-Long, Chen Chun, Wang Can, Wu Hao, Zhang Li-Jun, He Xiao-Fei. Music recommendation by unified hypergraph: Combining social media information and music content//Proceedings of the 18th ACM International Conference on Multimedia. Philadelphia, Pennsylvania, 2010: 391-400

[14] Mooney R J, Roy L. Content-based book recommending using learning for text categorization//Proceedings of the 5th ACM Conference on Digital Libraries. San Antonio, USA, 2010: 195-204

[15] Sarwar B M, Karypis G, Konstan J A, Riedl J. Item-based collaborative filtering recommendation algorithms//Proceedings of the 10th ACM International World Wide Web Conference. Hong Kong, China, 2001: 285-295

[16] Herlocker J L, Konstan J A, Terveen L G, Riedl J. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 2004, 22(1): 5-53

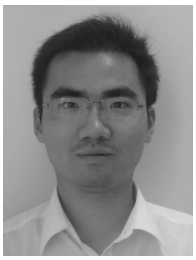
[17] Loannis K, Stathopoulos V, Jose J M. On social networks and collaborative recommendation//Proceedings of the 32nd ACM Special Interest Group on Information Retrieval. Boston, USA, 2009: 195-202

[18] Mahmood T, Ricci F. Towards learning user-adaptive state models in a conversational recommender system//Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems. Halle, Germany, 2007: 373-378

[19] Yoshii K, Goto M, Komatani K, Ogata T, Okuno H G. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences// Proceedings of the 7th International Conference on Music Information Retrieval. Victoria, BC, Canada, 2006: 296-301

[20] Basilico J, Hofmann T. Unifying collaborative and content-based filtering// Proceedings of the 21st International Conference on Machine Learning. Banff, Alberta, Canada, 2004: 65-72

[21] Cui Bin, Tung A K H, Zhang Ce, Zhao Zhe. Multiple feature fusion for social media applications// Proceedings of the ACM Special Interest Group on Management of Data. Indianapolis, Indiana, USA, 2010: 435-446



JIA Da-Wen, born in 1982, Ph. D. candidate. His research interests include recommender systems, Web data management.

ZENG Cheng, born in 1978, Ph. D., associate professor. His research interests include service computing and social computing.

PENG Zhi-Yong, born in 1963, Ph. D., professor and Ph. D. supervisor. His main research interests include complex data management, Web data management, and trusted data management.

CHENG Peng, born in 1989, M. S. candidate. His research interests include Web services, cloud computing.

YANG Zhi-Min, born in 1989, M. S. candidate. His research interests include Web services, cross-media information retrieval.

LU Zhou, born in 1990, M. S. candidate. His research interests include Web services, cloud computing.

Background

The volume of social media data have been increasing tremendously in the internet. Social media sharing and recommendation are meaningful and important. Most social media sites provide the group mechanism, where the user can manually create groups for media sharing. Users create and join groups for social purposes, and the formation of groups has gained great popularity and attracted enormous number of users. However, the existing group mechanisms in current social media sites have many disadvantages[5-8]. The most mentioned problem is that the groups in current social media sites are self-organized, which caused one topic may have a huge number of corresponding groups. It is a difficult task for users to either choose the right groups to join in by themselves or distribute a media object into appropriate group pools. Many researches focus on recommending groups for a given media object according to media content[6-8], and there is another line of research that focus on recommending groups to each user[4,9]. Though those group recommendation approach could recommend the best groups for users and media objects. It still does not satisfy the media sharing and recommendation task. Users with common interests may still separate in different groups, therefore, the media objects they sharing can not be seen by other group members. Considering from the viewpoint of social media recommendation, researches focus on the accuracy of recommendation algorithms. In social media environment, there exist immense amount of users and media objects, and the numbers are rapidly increasing every day. Although, the requirement of time efficiency in recommender system is not as crucial as that in retrieval system, in the circumstance of mass data and users, it is necessary to consider the factor of time efficiency for recommendation approaches. Obviously, group based media object recommendation would be much more efficient than user based media object recommendation. Furthermore, the computational complexity would be greatly reduced since the number of groups could be much less than the number of users.

To tackle these problems above, we propose a novel approach to discover groups automatically based on the preferences of users, called the Common Preference Group(CPG). We also propose a new social media data sharing and recommendation system architecture based on CPG and designed a CPG automatic mining algorithm. A series of experiments have been conducted to evaluate the different factors which affect algorithm performance based on randomly generated simulated data sets. Meanwhile, this paper compares our CPG mining algorithm with other algorithm which has similar functionality. The experimental results indicate that our algorithm could be applicable to real social media application with massive users.

This work is supported by the National Natural Science Foundation of China under Grant No. 61070011, Natural Science Foundation of Hubei Province in China, Wuhan Science and Technology Bureau in China under Grant No. 201150530139.