一种检测兴趣漂移的图结构推荐系统

叶红云,倪志伟,倪丽萍

(合肥工业大学 管理学院,合肥 230009)
E-mail: yehongyun@ hotmail. com

摘 要: 协同过滤是构造推荐系统最有效的方法之一. 其中,基于图结构推荐方法成为近来协同过滤的研究热点. 基于图结构的方法视用户和项为图的结点,并利用图理论去计算用户和项之间的相似度. 尽管人们对图结构推荐系统开展了很多的研究和应用,然而这些研究都认为用户的兴趣是保持不变的,所以不能够根据用户兴趣的相关变化做出合理推荐. 本文提出一种新的可以检测用户兴趣漂移的图结构推荐系统. 首先,设计了一个新的兴趣漂移检测方法,它可以有效地检测出用户兴趣在何时发生了哪种变化. 其次,根据用户的兴趣序列,对评分项进行加权并构造用户特征向量. 最后,整合二部投影与随机游走进行项推荐. 在标准数据集 MovieLens 上的测试表明算法优于两个图结构推荐方法和一个评分时间加权的协同过滤方法.

关 键 词: 图结构推荐; 兴趣漂移检测; 二部图投影; 随机游走
中图分类号: TP311 **文献标识码:** A **文 章 编 号:** 1000-1220(2012)04-0700-07

Novel Graph-based Recommender System with Interest Drift Detection

YE Hong-yun, NI Zhi-wei, NI Li-ping
(*School of Management*, *Hefei University of Technology*, *Hefei* 230009, *China*)

Abstract: Collaborative Filtering(CF) is regarded as one of the most successful approaches for building recommender systems. Among CFs, the study of graph-based recommendation methods has become a research hot spot. Graph-based methods consider users and items as vertices of a graph and leverage graphical theories to characterize the similarities between users and items. The recommendation list is built according to the similarities of candidate items with a given user. Though graph-based recommender systems have been widely studiedvand applied, all of them neglect an important fact that users´ interests usually change from time to time. Thus, they fail to model the dynamic interests of users reflected through user ratings. In this paper, we propose a novel graph-based recommender system with user interest drift detection. Firstly, we design a novel interest drift detection method that takes both the content variety of rated items and the change of a given user´s neighbors into consideration. This method is effective for capturing when and how a user´s interest changes. Secondly, rated items are weighted for constructing users´ feature vectors according to their interest drift series. Finally, we combine the bipartite graph projection and the random work approach for recommendation. The experiments on MovieLens data demonstrate that our recommender method outperforms two other graph-based recommendation approaches and a time-weighted memory-based collaborative filtering method.

Key words: graph-based recommendation; interest drift detection; bipartite graph projection; random walk

1 引 言

个性化推荐系统主要是根据用户历史行为及个人信息等构建用户的兴趣模型,以帮助用户快速过滤大量无用信息并预测满足其兴趣要求产品或对象[5]. 为了设计有效的推荐系统,人们已经研究了包括基于内容的推荐算法[12]、基于协同过滤的推荐算法[14],以及混合推荐算法[6]. 在这些方法中,基于协同过滤的推荐算法(collaborative filtering,简称 CF)由于其具有推荐效果良好,同时有具有较低的实现与维护代价而得到大量的研究和实际应用[1,7,16,23]. 近来,基于图结构的协同过滤算法成为新的研究热点[8,9,22-25]. 然而,当前基于图结构的算法都认为用户的兴趣是保持稳定的,忽略了用户兴趣的时刻变化. 因此,它们的推荐准确率往往有待提高.

众所周知,由于受到各种因素的影响,现实中用户的兴趣是随时间不断变化的,我们称用户兴趣发生了漂移. 例如,一位在校学生可能喜欢穿休闲服. 然而,当工作以后,他很有可能会对正装更感兴趣. 这种变化随时随地都在发生,如果推荐系统不能把握兴趣的变化,就很可能推荐过时的项给用户. 同时,用户的一些兴趣可能有较高频率的变化,而另一些兴趣则持续较长时间[7]. 因此,通过掌握用户的兴趣变化去提高推荐系统的性能是一项非常有研究价值的问题.

作为发现用户兴趣变化的一种途径,一些工作利用项的评分时间对项进行加权来反映用户兴趣的潜在变化. 这些方法的不足之处在于以评分时间为权重并能真正反映项的重要性,很久之前评分的项可能比用户近期评分的项更能反映该用户的兴趣. 例如,一个科幻爱好者可能评价过许多科幻电影以及少数一些剧情电影. 从他的整个评分历史来看,尽管一些科幻电影的评分时间可能比其它剧情电影早很多,但它们却更适合描述用户的喜好. 为了克服这种不足,一些学者利用基

收稿日期: 2010-11-25 基金项目: 国家"八六三"高技术研究发展计划项目(2007AA04Z116)资助; 国家自然科学基金项目(70871033)资助. 作者简介: 叶红云,女,1967 年生,博士研究生,主要研究方向为数据挖掘与智能管理; 倪志伟,男,1963 年生,教授,博士生导师,主要研究方向为数据挖掘、智能管理、决策支持系统等; 倪丽萍,女,1982 年生,博士研究生,主要研究方向为数据挖掘与智能管理.

于内容的信息去检测用户兴趣漂移. 然而,在许多情况下可以利用的项信息并不足以区分两个不同类别的项. 此时仅利用基于内容的信息不能正确反映出用户兴趣的漂移. 幸运的是,喜爱不同兴趣的项的用户组也往往不同. 所以,当用户的兴趣发生变化时,与他(她)有相似喜好的用户组也会同时发生变化. 因此,通过将基于协同的信息与基于内容的信息结合可以更准确地识别兴趣的变化. 然而,当前的研究都没有尝试利用协同信息去挖掘用户的兴趣变化.

本文提出一种新的兴趣漂移检测方法,它同时考虑了基于内容和基于协同的信息. 除此以外,本文还提出一个新的图结构推荐算法 BPIR(bipartite graph projection and interest weighted ranking),该算法包含三个步骤. 首先,利用用户已评分项的内容信息及用户邻居用户的变化去检测用户的兴趣是否漂移. 这里邻居用户是指通过观察 co-rating 所发现的那些与给定用户兴趣相似的用户. 户与项之间的关系用二部图进行描述,其中,二部图的一类顶点表示用户,另一类顶点表示项. 为了获得表示项之间相似关系图,将二部图进行一维投影. 最后,兴趣加权的随机游走被用来计算候选项与给定用户间的相似度,通过对候选项进行排序就可以产生推荐序列. 本文的主要贡献在于:

· 在图结构推荐系统中解决用户兴趣漂移问题. 为推荐系统尤其是图结构推荐系统的研究提供一种新的思路.

· 提出了一种新颖的用户兴趣漂移检测方法. 它将项内容的变化和用户的邻居用户的变化融合到一起. 该方法通过潜在兴趣而不是时间对项进行加权.

· 通过在真实数据集上的实验,发现了一些有趣的现象. 通过分析,我们揭示了它们背后隐藏的规律.

2　相关工作

2.1　图结构协同过滤

自上世纪 90 年代以来,人们提出或实现了大量基于协同过滤的推荐系统,其中最为常见的有两类 CF 算法: Model-based 方法和 Memory-based 方法. Model-based 方法,如基于贝叶斯网络的方法[11]、基于最大熵的方法[21]等,利用已有的用户评价数据建立一个模型,然后根据此模型进行评价预测. 对于一个用户而言,该方法对项的排序是综合考虑与其相似的用户对项的评分值,以及同相似用户的相似程度来实现的[10,18].

近来,有学者提出基于图中顶点相似计算的方法来进行协同推荐[8,9,17,22-25],本文把它们归纳为图结构协同过滤. 在这些方法中,用户和项被当成图中的结点,每一个用户-项对之间的关系被它们在图中的相似情况所描述. 通过揭示侯选项与给定用户间的相似度来产生推荐列表. 图结构 CF 由两个子问题组成,即建立何种关联图以及如何揭示用户-项对间的相似度.

目前主要有两类关联图: 由用户和项组成的关联图[8,9]、仅由项组成的关联图[22-25]. 关联图不同,所用的相似判断方法也不同.

引文[8,9]把用户和项放入同一个关联图中并进行项推荐. 随机游走是他们所用的相似判断方法之一,并利用 ACT (average commute time)作为评价标准. 然而,ACT 对于图的远端非常依赖[15]. 因此,利用 ACT 计算出来的相似度通常与实际相似度有较大的误差. 除了 ACT 以外,引文[8,9]还测试了其它几种相似评判方法,用来对用户进行项推荐. 在进行推荐时,引文[8,9]所提出的算法可以覆盖所有的用户和项,然而,把所有用户和项用一个图表示会大幅提高此类算法的时空复杂度.

利用不同项在同一用户评分序列中的共同出现次数,引文[22,24]建立项-项关联图. 引文[23]更是利用余弦相似度计算项间的关联. 但是这些方法的共同局限性在于: 每个评价过项 i 和项 j 的用户会对 i,j 的相似度产生相同的贡献. 假设用户 A 仅评价过项 i 和 j,而用户 B 除了 i,j 之外还评价过其它 100 个项. 在这种情况下,我们有理由相信用户 A 应当比用户 B 更能说明 i 与 j 相似. 引文[25]利用二部图去描述用户与项之间的关系. 通过一维投影,将二部图转化为项之间的关联图. 因为此算法利用资源分配的方法对二部图进行投影使得两个项有非对称的相互推荐能力,其表现也优于一般的协同过滤算法.

然而,引文[25]在推荐时仅考虑项之间的直接联系. 例如,如果一个项 i 从来没有和给定用户评价过的项 co-rating 过,则项 i 与给定用户间的相似度就是 0. 但实际上,如果存在许多既与项 i 相似又与给定用户评价过的项相似的项,则给定用户也很有可能对项 i 产生兴趣. 本文将揭示引文[25]所提的算法在关联图比较密集的时候会有很好的表现,却不适合用于关联图稀疏的情况.

在计算节点相似度的时候,一种抵消 ACT 远端依赖的方法是每次游走有限步[25]. 另一种方法是周期性"重启动"节点 i 到 j 的游走,即在每一游走步以概率 c 返回节点 i 并重新开始游走,如此将使得图中偏远部分很难有机会被走到[17]. 在网页排序的 PageRank 算法中,也采用了随机重启的思想. 此外,还有研究多媒体对象中各媒体属性间的关联关系发现[20]、情感与音乐属性的关联从而进行音乐推荐[13]的工作中也采用了带重启动的随机游走(RWR,random walk with restart).

2.2　兴趣漂移检测

实际应用中,用户的兴趣和需求会随时间不断发生变化. 识别出用户的兴趣在何时何地发生了何种变化,对提高推荐系统的性能有重要意义[19],尤其是当系统运行了较长的时间以后.

尽管绝大多数基于协同过滤的推荐系统潜在地认为用户兴趣是稳定不变的,并没有考虑到此类变化. 然而,此问题已经引起了越来越多的注意和研究[5,7,10,18,19]. 其中,引文[5]利用遗忘机理提出了一个模拟 blogger 的兴趣模型,并利用两个模型分别模拟长期和短期兴趣. 引文[7]则利用一个衰减因子计算不同项的时间权重并赋予过时的数据较低的权重. 引文[10]通过定义一个时间加权的函数并利用时间加权项选择邻居用户来提升传统的协同过滤算法性能. 引文[18]提出一个混合模型并把引文[7]所定义的时间敏感函数嵌入到推荐过程中. 引文[19]利用聚类模型和自相似模型检测用户随时间变化的模式,并实现动态和静态两种推荐.

尽管实验显示以上方法可以比传统的协同过滤产生更好

的推荐效果,但它们有个共同的不足,即以上方法仅利用时间因素对项进行加权,并给过时的数据更低的权重. 然而,一些过时的评分可能会比新近的评分更重要. 为此,本文将通过考虑兴趣的时间顺序来解决此不足.

此外,到目前为止所有的兴趣漂移检测方法都被用来提升传统的 memory-based 协同过滤,还没有图结构协同过滤算法引入兴趣漂移检测问题.

3 BPIR 算法描述

3.1 算法框架

BPIR 算法由三部分组成. 在第一部分将利用兴趣漂移检测算法对用户的评分项进行加权并创建用户特征向量. 在第二部分我们利用二部图投影和资源分配机制建立项之间的关联图. 在最后一部分,通过将给定用户的兴趣加权特征向量在项关联图上进行带重启动的随机游走(RWR),将候选项进行排序并产生推荐序列. 图 1 显示了 BPIR 算法的框架.

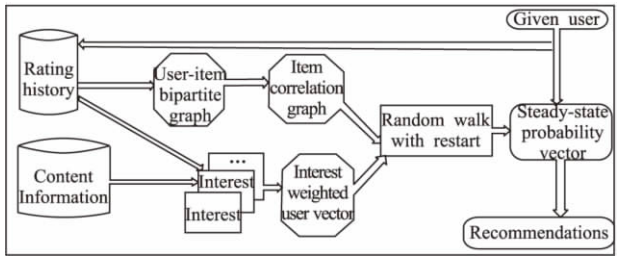

图 1 BPIR 算法框架图

Fig. 1 Framework of BPIR algorithm

接下来将详细介绍 BPIR 算法,简便起见,定义 $U=\{U_1, U_2, \cdots, U_m\}$ 为用户集合,$I=\{I_1, I_2, \cdots, I_n\}$ 为项集合.

3.2 兴趣漂移检测

直观地,当用户的兴趣发生变化时,他(她)所喜爱项的特征会随之变化. 以电影为例,如果给定用户所喜爱的电影从动作片转移到爱情片,则其兴趣即发生了漂移. 然而,很多情况下,仅利用电影描述信息很难反映出用户的兴趣变化. 幸运的是,利用协同信息可以优化分类. 因为,当给定用户的兴趣发生变化时,与其相似的邻居集也会发生变化,因此,可以将评分项的内容变化与用户邻居集的变化结合起来表示评分项所代表的不同兴趣,并以此检测用户的兴趣是否发生了漂移.

本文将兴趣归纳为一些相似项的组合. 我们将一个兴趣中最近评分项的评分时间称为此兴趣的激活时间. 所有的兴趣将按其激活时间排序,排序以后的索引称为兴趣索引. 一个兴趣中最早评分的项被称为此兴趣的首项. 当计算两个项的相似度时,我们同时考虑基于内容与协同的信息. 特别地,给定项 I_i 与 I_o,我们将它们映射到一个二维空间中,如图 2 所示. 横轴坐标代表项之间的内容相似度,而纵轴表示它们的协同相似度. 如果 I_i 与 I_o 之间的欧氏距离不大于阈值 r(这意味着 I_i 在图 2 中的阴影部分),则认为它们相似并位于同一个兴趣中,否则我们向前继续寻找与 I_i 相配的兴趣.

我们定义内容相似为 δ_H,协同相似为 δ_V,公式如下:

$$\delta_H(I_i, I_o)=\frac{|G_{I_i}\cap G_{I_o}|}{\min(|G_{I_i}|, |G_{I_o}|)}, \delta_V(I_i, I_o)=\frac{|\Gamma_{I_i}\cap \Gamma_{I_o}|}{\min(|\Gamma_{I_i}|, |\Gamma_{I_o}|)}$$

其中,G_{Ii} 表示 I_i 的内容特征(例如,本文采用电影的风格为内容特征),Γ_{Ii} 代表对 I_i 评分的用户集合.

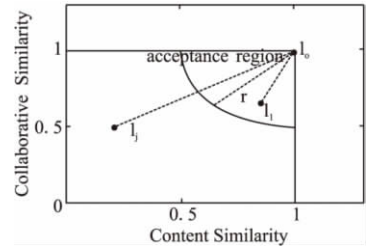

图 2 I_i 位于接收域,所以它属于以 I_o 为首项的兴趣,而 I_j 则不属于此兴趣

Fig. 2 I_i is in the acceptance region, so it belongs to the interest whose first item is I_o, while I_j does not

如图 2 所示,我们将每个兴趣的首项 I_o 映射到(1,1)点,并定义欧氏空间如下:

$$ED(I_i, I_o)=\sqrt{(1-\delta_H(I_i, I_o))^2+(1-\delta_V(I_i, I_o))^2}$$

本文提出的兴趣漂移检测算法流程如图 3 所示. O_j 是给定用户以 j 为索引的兴趣的首项,Interest_Num 是给定用户的兴趣数.

```
Input: Q is item queue sorted by items´ rating time;
       r is the threshold
Output: each item´s interest Index

Interest_Num = 1; Current_Item = pop(Q);
Initialize a new interest, put Current_Item as O_Interest_Num;
while Current_Item do
    Flag = 0;
    If ED(Current_Item, O_Interest_Num) > r then
        For j (Interest_Num - 1) to 1 do
            if ED(Current_Item, O_j) ≤ r then
                Change interest Index;
                /* Interest j becomes current interest. */
                Flag = 1; break;
        if Flag = = 0 then
            Initialize a new interest;
            Interest_Num + +;
            Put Current_Item as O_Interest_Num;
    Interest_Index(Current_Item) = Interest_Num;
    Current_Item = pop(Q);
Return each item´s interest Index
```

图 3 算法 Interest_Drift_Detection 流程

Fig. 3 Flow chart of Interest_Drift_Detection algorithm

我们以 Movielens 数据集(数据细节将在第 4.1 节描述)中用户 1 的前 50% 评价为例. 利用兴趣漂移检测算法所得的部分结果如表 1(见下页)所示. 根据基于内容的相似,电影被准确地分入不同的兴趣中. 根据基于协同的信息,尽管兴趣 1 和 5 中的电影拥有相同的风格,它们仍被分入了不同的兴趣.

如果仔细观察这些电影,我们会发现兴趣 1 中的电影都拍摄于 1990s,而兴趣 5 中的电影都拍摄于 1980s. 同时,在 IMDB[1] 中它们的平均评分也相差甚远. 电影 264 与 260 的分值都很低,但电影 89 的分值却很高. 这些不同使得两类电影被不同的用户群所喜欢,同时,不同的用户群也反映了两类电影的差别. 因此,我们的方法可以利用项之间的协同信息正确实现电影的分类. 根据前面的讨论,有时评分时间不能准确反映项的重要性. 因此,我们根据 U_q 评分序列中 I_i 所属兴趣的索引对 I_i 进行加权:

$$F(I_i) = e^{\text{Interest\_Index}(I_i) - Interest\_Num_{U_q}}$$

$F(I_i)$ 的值位于区间 $[e^{1-Interest\_Num_{U_q}}, 1]$. 指数函数被广泛应用于时间加权研究中[5,7,18],主要基于两个原因:

1. 如果 $\text{Interest\_Index}(I_i) > \text{Interest\_Index}(I_j)$,则 $F(I_i) > F(I_j)$

2. 如果 $\text{Interest\_Index}(I_i) > \text{Interest\_Index}(I_j)$,则 $F'(I_i) > F'(I_j)$

$F'(I_i)$ 是 $F(I_i)$ 的一阶导数. 这两个性质使得我们的加权函数与心理学家所广泛接受的两个观点相一致. 首先,与记忆类似,人类的兴趣随着时间的推移而消减. 其次,随时间推移,此消减速度会逐渐变慢直至稳定.

表 1　当 r = 0.6 时,用户 1 的兴趣漂移检测结果

Table 1　Interest_Drift_Detection results for user 1 with r = 0.6

Interest_Index	I_o	Movie_ID	Genre
1	264	264,260	Sci-Fi, Thriller
2	248	248,249,251	Comedy
3	114	114	Animation
4	14	14,126,237,60	Drama
5	89	89	Sci-Fi, Thriller
6	48	48	Documentary

3.3　关联图

设 U 表示用户集合,I 表示项集合,可以用二部图 G = < X,E > 表示用户与项之间的关系,其中 X 是由 U 和 I 所有元素构成的顶点集,表示用户的顶点构成二部图的一类顶点,表示项的顶点构成二部图的另一类顶点. 如果顶点 Uq 与顶点 Ij 间存在一条连边,则意味着用户 Uq 对项 Ij 进行了评价. 图 4 即为一个用户-项二部图实例.

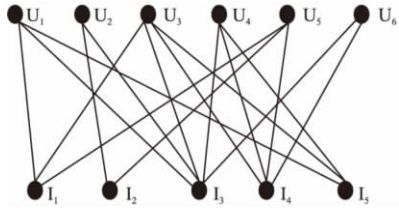

图 4　用户-项二部图实例

Fig. 4　An example of user-item bipartite graph

为了尽可能反映原二部图中隐藏的信息,本文使用引文[25]所提出的基于资源分配方法的二部图投影方法. 该方法首先给 I 中的结点赋予相同数目的资源(例如,该资源可以被引申为推荐能力). 然后,I 每个结点中的资源被平均地分配给其在 U 中的邻居. 最后,每个 U 结点获得的资源会再次被平均返回给每个 I 邻居. 通过投影,从结点 I_i 流向其它 I 结点的资源数就反映了其它结点与 I_i 的相似度,也同时意味着 I_i 推荐其它项的能力.

投影过后,二部图被转化成项之间的关联图. 在此关联图中,$s(I_i, I_j)$ 表示连接结点 I_i 与 I_j 的有向边权重,它可以被用来衡量 I_i 与 I_j 的相似度. 构造矩阵 C 满足 $C_{i,j} = s(I_i, I_j)$. 通过归一化,C 即成为关联图的关联矩阵.

因此,在投影的过程中资源被分配了两次,本文称此投影过程为 BFS(back and forth similarity). 根据引文[25]的思想,我们归纳投影和创建关联矩阵 C 的过程为算法 BFS_C,如图 5 所示.

1. Construct matrix W_{n*m} to represent G. $W_{i,j} = 1$, if U_j rated I_i in G, otherwise $W_{i,j} = 0$;

2. Construct matrix W′, W″, with

$$W'_{i,j} = \frac{W_{i,j}}{\sum_{k=1}^{m} W_{i,k}}, W''_{i,j} = \frac{W'_{i,j}}{\sum_{k=1}^{n} W_{k,j}}$$

3. Construct matrix C, if $i \neq j$ then $C_{i,j} = s(I_i, I_j) = \sum_{k=1}^{m} W_{i,k} * W_{j,k}''$, else $C_{i,j} = 0$;

4. Normalize C with the sum of entries in each column to 1;

5. Return C

图 5　BFS_C 算法流程

Fig. 5　Flow chart of algorithm BFS_C

在关联矩阵中,$C_{i,j}$ 是从 I_j 的视角来看,I_i 与 I_j 之间的亲密度. 就随机游走而言,$C_{i,j}$ 是当前状态从 I_j 跳跃到 I_i 的概率. 因此 C 是一个状态转移矩阵,其中每个元素代表相应结点对之间的转移概率. 例如,图 4 的关联阵 C 即为:

$$C = \begin{pmatrix} 0 & 0.286 & 0.189 & 0.226 & 0.280 \\ 0.160 & 0 & 0.162 & 0.129 & 0 \\ 0.280 & 0.429 & 0 & 0.419 & 0.440 \\ 0.280 & 0.286 & 0.351 & 0 & 0.280 \\ 0.280 & 0 & 0.297 & 0.226 & 0 \end{pmatrix}$$

3.4　执行推荐

给定一个用户 U_q,可以用一个向量 V'_{Uq} 来表示他的兴趣模型,向量中的每个元素 $V'_{Uq}[i]$ 代表一个项:

$$V'_{U_q}[i] = \begin{cases} F(I_j), & if\ U_q\ rated\ I_j \\ 0, & otherwise \end{cases} \quad (1)$$

对 Uq 的兴趣模型 V'_{Uq} 归一化后得到特征向量 V_{Uq}. V_{Uq} 代表用户 Uq 随机游走初始概率分布的向量. 接下来,V_{Uq} 在关联图上进行带重启动的随机游走.

考虑从 V_{Uq} 开始的随机游走,作为对人际关系的模拟,每一步它要保证两个性质: 首先,如果项 I_i 与 U_q 是紧密联系的,那么以 I_j 为纽带,U_q 与 I_i 之间也会建立一定联系. 其次,因为此联系是间接建立的,相比于 U_q 与 I_j 之间的联系,U_q 与 I_i 之间的关系会有一定的“衰减”. 当随机游走到达 I_i 的时

[1] The Internet Movie Database, URL: http://akas.imdb.com/.

候,它通过随机选择 I_i 的一个邻居并继续游走来保证此两个性质. 除此之外,在做出决定之前,随机游走以一定的概率返回到初始结点,以此抵消对图远端的依赖[15].

用概率向量 $S_{Uq}=(S_{Uq}[1],S_{Uq}[2],\cdots,S_{Uq}[n])^T$ 表示由随机游走生成的推荐列表. 其中,$S_{Uq}[i]$ 表示从 V_{Uq} 开始的随机游走最终停留在结点 I_i 的概率,$S_{Uq}[i]$ 代表了 I_i 与 U_q 的亲密度: $S_{Uq}[i]$ 越大,U_q 与 I_i 关系越紧密. 因此,通过对 S_{Uq} 排序,很容易很到 U_q 的 top-K 个推荐项.

Input: G is the bipartite graph;
U_q is the user that we recommend for;
c is restart probability for RWR;
r is acceptance region's radius
Output: a recommendation list for U_q
1. Get U_q's rating queue Q;
2. Interest_Drift_Detection(Q; r)
3. Matrix C = BFS _C(G);
4. Get normalized vector V_{Uq} from Equation 1;
5. t = 0; $S_{Uq}=V_{Uq}$;
6. While(S_{Uq} doesn't converge || t < MAX_LOOP) do
7. $S_{Uq}=(1-c)*C*S_{Uq}+c*V_{Uq}$;
8. t + + ;
9. Descending sort S_{Uq} and generate recommendation list for U_q;
10. Return the recommendation list

图 6 BPIR 算法流程

Fig. 6 Flow chart of algorithm BPIR

推荐算法 BPIR 的总体流程如图 6 所示. BPIR 为每个用户运行带重启动的随机游走,c 表示随机游走的重启动概率,(1-c) 是衰减因子. 在 4.5 节,我们将展示一些 c 的取值与关联图的统计信息之间的关系.

表 2 当 c = 0.5 时,为用户 U_2 运行 BPIR 的结果

Table 2 BPIR results for user U_2 with c = 0.5

	I_1	I_2	I_3	I_4	I_5
S_U	0.1054	0.2976	0.3758	0.1353	0.0859
Predicted Rank	2	\	\	1	3

为了更好地解释 BPIR 算法,考虑在 3.3 节提出的简单例子. 如果 I_2,I_3 属于同一个兴趣,那么 $V_U=(0,0.5,0.5,0,0)^T$ 是 U_2 的用户向量. 运行 BPIR,最终得到的稳态概率向量 S_U 以及排序后的推荐列表如表 2 所示.

4 实 验

实验采用当前推荐系统研究中的一个标准数据集 MovieLens 数据集. 通过与三种对比方法的实验对 BPIR 算法的有效性进行了验证.

4.1 数据集和实验设计

GroupLens 小组从一个进行电影推荐的网站 MovieLens[1] 中整理出了一个标准数据集[2]. 本文使用了其中打分超过 20 部电影的用户,其中共包含约 100,000 个评分,涉及到 943 个用户,以及 1682 部电影.

对于每个用户的评分序列,我们以一定的比例对它们进行分割并以前一部分为训练集而后一部分为测试集. 在本实验中我们一共构造了四对训练与测试数据对. 在前三个数据对中,每个用户的前 50%,70%,90% 评分分别被用来做训练数据,剩余的评分则用做测试集. 在最后一个数据对中,每个用户的前 80% 评分被用做训练数据而接下来的两个评分被用做测试集. 80%-2 数据对可以用来测试当训练数据充足并且系统只需要预测用户在接下来的极短时间内最有可能的评分项时的情况. 同时,80%-2 数据对是 9%-1% 数据对的模拟,因为许多用户的评分小于 100 个,所以采用 99%-1% 数据会使他们的测试数据集为空.

为验证BPIR的有效性,以两种图结构推荐算法ItemRank[23] 和 NBI[24] 以及一个 memory-based 协同过滤算法(CFID)[10] 为对比算法. 其中,CFID 考虑了用户的兴趣漂移问题.

4.2 评价标准

本文采用 DOA(Degree of Agreement) 和 HR(Hit Ratio) 为评价标准. DOA 衡量正确排序的项对在所有项对中所占的比例,它已经被大量应用到图结构算法中[8,9,16,22,24]. DOA 的定义如下.

对于用户 U_q,设 L_{Uq} 表示训练集中被其评价的项,T_{Uq} 表示测试集中被其评价的项,那么,$NW_{Uq}=I-(L_{Uq}-T_{Uq})$ 则表示所有未出现在其训练集及测试集中的项.

定义 check_order 函数如下:

$$check\_order_{Ui}(I_j,I_k)=\begin{cases}1, if(predict\_rank_{ij}\geqslant prdict\_rank_{ik})\\0, otherwise\end{cases}$$

PR_{ij} 是推荐列表中对 I_j 的预测位置. 给定用户 U_q 的 DOA 得分定义如下:

$$DOA_{Uq}=\frac{\sum_{j\in T_{Uq},k\in NW_{Uq}} check\_order_{Uq}(I_j,I_k)}{|T_{Uq}|*|NW_{Uq}|}$$

随机预测的 DOA 值大约为 50%,DOA 值为 100% 则意味着所有项对均排序正确. 实验中,DOA 的总体评价效果取所有用户 DOA 值的平均.

HR 衡量预测中用户测试数据在用户测试数据中的比例[23]. 当 HR 取到 100% 时意味着所有的用户测试数据都被准确预测. 此方法如下定义:

$$HR_{Uq}=\frac{\#hits}{|T_{Uq}|},HR=\frac{\sum HR_{Uq}}{|U|}$$

4.3 实验结果

对于每一种算法,本文通过优化选择它们的最优参数进行比较. 对于 ItemRank,衰减因子 α 被设置为 0.01,对于 CFID 我们设置 $\lambda=1$,N = 50. 表 3(见下页) 显示了不同方法在各个数据对上的表现情况.

通过图 3 可以看出 BPIR 算法明显优于其它三个算法. 在 DOA 标准下,训练评分数据越多,BPIR 算法的优势越明显,例如 90%-10% 数据对和 80%-2 数据对. 使用 HR 为标准时,在各个数据集上的改进效果更为明显.

另一个有趣的现象是 NBI 算法在 DOA 标准下表现优于 ItemRank. 在实验中,ItemRank 与 NBI 的最大不同在于它们构造关联图的方式. 这说明 NBI 算法使用的关联图比 Item-

Rank 的关联图更有效. 这印证了本文的一个假设: 利用资源分配方法进行的二部图投影比项的 co-rating 次数更能揭示项的相似性. 然而,由于 NBI 算法推荐时的覆盖率不足,它在 HR 标准下效果比较差.

表 3　不同方法对比结果

Table 3　Performance comparison among different algorithms

(a. DOA in %)

Alg. \Split	50%-50%	70%-30%	90%-10%	80%-2
ItemRank	84.8017	84.8731	84.5663	86.4693
NBI	85.3381	85.8135	86.2118	87.4497
CFID	81.2974	82.4131	82.5076	84.7521
BPIR	**86.0399**	**86.7138**	**87.5113**	**89.2867**

(b. HR in %)

Alg. \Split	50%-50%	70%-30%	90%-10%	80%-2
ItemRank	23.2444	16.7986	8.2020	3.2344
NBI	22.2228	16.3866	7.7735	2.5451
CFID	22.2358	16.2647	8.1348	3.8176
BPIR	**24.6532**	**19.9415**	**10.7301**	**4.2418**

4.4　接收域半径

当检测兴趣漂移时需要为接收域使用最优的半径 r. 表 4 显示了使用不同 r 得到的实验结果.

表 4　不同 r 的 BPIR 对比结果(DOA %)

Table 4　BPIR results for different r (DOA in %)

r\Split	50%-50%	70%-30%	90%-10%	80%-2
0	85.9224	86.7138	87.5113	89.2867
0.1	85.9173	86.6720	87.4417	89.2202
0.2	85.9419	86.6328	87.4390	89.1759
0.3	85.9530	86.6492	87.3487	87.0785
0.4	85.9853	86.6346	87.2538	89.0154
0.5	86.0004	86.5641	87.1882	88.8674
0.6	86.0399	86.5094	87.0278	88.6267
0.7	85.9766	86.4788	86.7511	88.5176
0.8	85.8366	86.3597	86.7069	88.2968
0.9	85.6928	86.1939	86.5640	88.0410
1.0	85.4938	86.0795	86.5099	87.8443

由于 r 控制着兴趣的产生和大小,它决定着算法的表现效果,同时,对于不同的数据集,其最优的r也不同. 为了进一步研究 r 在不同情况下的表现情况,我们将 Movielens 数据集以不同比例划分成更多数据对. 图 7 显示了每个数据对的最优 r 值. 可以看出,在训练集从 10% 变化到 60% 的过程中,其最优 r 值保持在 0.4 到 0.7 之间. 然而,最后三个数据对的最优r值为0,这意味着此时每个项组成一个单独的兴趣,本文的兴趣漂移检测算法退化成一个时间加权方法. 此时,那些很久以前评分的项可以被忽略,用户的兴趣可以由他近期的评分反映. 这个结果说明当用户刚在一个推荐系统中注册时,受到好奇心等的驱动,他的兴趣会经常发生变化. 随着时间的推移,当用户已经对大多数类别的项评分以后,他的兴趣会逐渐稳定,短期的评分就可以反映他的兴趣.

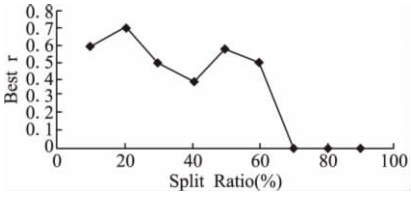

图 7　不同大小训练集的最优 r 值

Fig. 7　The best r for different size of training sets

4.5　重启动概率

在 PageRank 算法中重启动概率 c 经常取为 0.15,而在实验中发现,在 c 取 0.99 时 BPIR 算法推荐效果最好. 这意味着对于此四个数据对而言单步随机游走比多步随机游走更合适.

根据引文 [17] 的分析,Movielens 是一个比真实数据更稠密的数据集. 我们在不同的数据对中观察给定不同 c 的情况下随机游走的表现情况. 图 8 显示了最优 c,λ(λ 的定义请参见引文 [17]) 与关联图中的非零元素比例之间的关系. 通过图 8 可以看出,关联图越稠密,λ 值越小而最优 c 值越大.

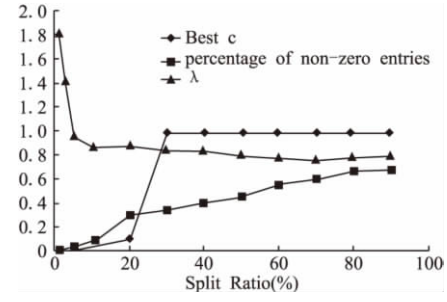

图 8　最优 c,λ 与关联图中的非零元素比例之间的关系

Fig. 8 The relationship among best c, λ and the non-zero entry percentage of each correlation graph

一方面,当数据足够稠密,电影之间有足够的直接关联,此时,基本上不需要考虑间接相似的贡献,带重启动的随机游走退化成为单步随机游走. 另一方面,当数据非常稀疏,随机游走不需要经常重新启动因为结点之间缺少足够的直接联系. 此时,考虑间接相似的多步随机游走要优于单步随机游走. 这也验证了本文 2.1 节的另一个观点: 由引文 [25] 提出的算法 NBI 适合于稠密的关联图,而不适合用于数据稀疏的情况.

引文 [14] 指出现实网络的 λ 值一般位于 2 与 3.5 之间,在本文实验中相应的 c 值位于 [0,0.05] 区间. 由于现实网络非常稀疏,因此重启动概率比较小,此时需要利用多步随机游走揭示项之间的关系. 相似的,引文 [23] 通过观察发现基于随机游走的方法可以在直接联系不足时更好地描述项之间的关系,认为基于随机游走的相似计算方法更适合于稀疏数据.

5　结论与展望

本文展示了一种考虑用户兴趣漂移的图结构推荐算法. 该算法首先映射每个用户的评分序列到兴趣序列,并通过兴趣序列对评分项加权. 本文提出的兴趣漂移检测算法有两个优良特点: 首先,它不仅考虑了基于内容的项相似而且利用了项之间的协同信息. 其次,本方法可以发现用户兴趣在何时发生了何种变化,并且利用兴趣对项加权. 再者,基于资源分配

的二部图投影算法被用来构建项之间的关联图. 最后,利用兴趣加权的评分项构造用户的特征向量并通过在项关联图上进行带重启动的随机游走产生推荐序列. 通过在 Movielens 数据集上与其它一些优秀算法的比较证实了算法的有效性. 本文揭示了重启动概率与关联图的稠密度之间的关系. 另外,本文还发现了几个有趣的现象,相信会对其它研究基于随机游走进行推荐的学者带来一些启示.

接下来,我们打算进一步研究用户的兴趣漂移问题,并期待提出更有效的推荐算法. 此外,还有许多有趣问题值得进一步研究,例如如何检测与利用用户的兴趣漂移模式、如何避免噪声兴趣的负面影响等等. 同时,通过本文我们发现在进入系统一定时间以后用户的兴趣可以由他的短期评分项反映. 检测何时会发生这种变化以及此时用户的兴趣究竟由多少项决定等都是未来工作的努力方向.

References:

[1] Movielens [EB/OL]. http://www.movielens.umn.edu, 2003.

[2] Movielens datasets [EB/OL]. http://www.grouplens.org/node/73#attachments, 2007.

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

[4] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine [J]. Computer Networks and ISDN Systems, 1998, 30(1-7): 107-117.

[5] Cheng Y, Qiu G, Bu J, et al. Model bloggers' interests based on forgetting mechanism [C]. In Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008: 1129-1130.

[6] Debnath S, Ganguly N, Mitra P. Feature weighting in content based recommendation system using social network analysis [C]. In Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008: 1041-1042.

[7] Ding Y, Li X. Time weight collaborative filtering [C]. In CIKM'05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, New York, USA, 2005: 485-492.

[8] Fouss F, Pirotte A, Renders J-M, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation [J]. IEEE Transaction on Knowledge and Data Engineering, 2007, 19(3): 355-369.

[9] Fouss F, Pirotte A, Saerens M. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation [C]. In 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005), 19-22 September, Compiegne, France, IEEE Computer Society, 2005: 550-556.

[10] Gong S, Cheng G. Mining user interest change for improving collaborative filtering [A]. In Proceedings of the Workshop on Intelligent Information Technology Application (IITA'08) [C], IEEE Computer Society, 2008: 24-27.

[11] Breese C K J, Heckerman D. Empirical analysis of predictive algorithms for collaborative filtering [C]. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998.

[12] Kim J W, Lee B H, Shaw M J, et al. Application of decision-tree induction techniques to personalized advertisements on internet storefronts [J]. International Journal of Electronic Commerce, 2001, 5(3): 45-62.

[13] Kuo F-F, Chiang M-F, Shan M-K, et al. Emotion-based music recommendation by association discovery from film music [C]. In MULTIMEDIA'05: Proceedings of the 13th Annual ACM International Conference on Multimedia, New York, NY, USA, 2005: 507-510.

[14] Latapy M, Magnien C, Vecchio N D. Basic notions for the analysis of large two-mode networks [C]. Social Networks, 2008: 31-48.

[15] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks [J]. Journal of the American Society for Information Science and Technology, 2007, 58(7): 1019-1031.

[16] Linden G, Smith B, York J. Amazon.com recommendations: item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-80.

[17] Liu Qi, Chen En-hong. Collaborative filtering through combining bipartite graph projection and ranking [J]. Journal of Chinese Computer Systems, 2010, 31(5): 835-839.

[18] Ma S, Li X, Ding Y, et al. A recommender system with interest-drifting [C]. In Web Information Systems Engineering-WISE 2007, 8th International Conference on Web Information Systems Engineering, Nancy, France, Lecture Notes in Computer Science, Springer, 2007, 4831: 633-642.

[19] Min S, Han I. Detection of the customer time-variant pattern for improving recommender systems [J]. Expert System with Application, 2005, 28(2): 189-199.

[20] Pan J-Y, Yang H-J, Faloutsos C, et al. Automatic multimedia cross-modal correlation discovery [C]. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, ACM, 2004: 653-658.

[21] Pavlov D, Pennock D M. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains [A]. In Advances in Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada [C], MIT Press, 2002: 1441-1448.

[22] Pucci A, Gori, M, Maggini M. A random-walk based scoring algorithm applied to recommender engines [C]. In Advances in Web Mining and Web Usage Analysis, 8th International Workshop on Knowledge Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, Revised Papers, 2006: 127-146.

[23] Yildirim H, Krishnamoorthy M S. A random walk method for alleviating the sparsity problem in collaborative filtering [C]. In RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems, Ney York, NY, USA, 2008: 131-138.

[24] Zhang L, Zhang K, Li C. A topical pagerank based algorithm for recommender systems [C]. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008: 713-714.

[25] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation [C]. Physical Review E, 2007: 46-115.

附中文参考文献:

[17] 刘 淇,陈恩红. 结合二部图投影与排序的协同过滤 [J]. 小型微型计算机系统,2010,31(5):835-839.