结合二部图投影与排序的协同过滤

刘 淇, 陈恩红

(中国科学技术大学 计算机科学与技术学院, 安徽 合肥 230027)
E-mail: chenenh@ustc.edu.cn

摘 要: 协同过滤是推荐系统中应用最为广泛的方法. 提出一类基于二部图一维投影与排序相结合的协同过滤算法, 文中采用结构相似进行二部图投影并利用随机游走对节点排序. 该方法不仅可以防止冷启动, 具有较高准确度, 且可扩展性良好. 另外, 该算法可以避免低覆盖率造成的推荐不准确. 算法可以有两类不同的实现, 分别是基于项协同过滤的项排序算法和基于用户协同过滤的用户排序算法, 在标准数据集 MovieLens上的测试表明了算法的有效性.

关 键 词: 协同过滤; 二部图投影; 结构相似; 排序; 随机游走
中图分类号: TP311 文献标识码: A 文 章 编 号: 1000-1220(2010)05-0835-05

Collaborative Filtering Through Combining Bipartite Graph Projection and Ranking

LIU Qi, CHEN En-hong
(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Collaborative Filtering is the most widely used approach in recommender systems. This paper proposes a novel collaborative filtering approach through combining bipartite graph projection and ranking. In our approach bipartite graph is projected based on structural similarity and random walk is used to rank the nodes. This method can not only deal with "cold start problem" to get high precision but also have good scalability. Moreover, our method can generate the rank based on the similarity between all user-item pairs, thus making it avoids inaccuracy caused by low coverage rate. The algorithm is tested in both the item-collaborative-based item ranking way and the user-collaborative-based user ranking way. The experimental results obtained on a benchmark dataset MovieLens clearly show the effectiveness of our proposed approach.

Key words: collaborative filtering; bipartite graph projection; structural similarity; ranking; random walk

1 引 言

推荐系统根据用户与系统的交互历史以及用户的个人信息等构建用户的兴趣模型, 去预测用户可能感兴趣的产品或项. 推荐系统大致可以分为三类[1]: 基于协同过滤的方法(collaborative filtering)[2-4], 基于内容的方法(content-based filtering)[5-8]和将二者结合的混合推荐算法(hybrid recommendation)[6,7]. 其中, 基于协同过滤(CF)的方法在保持较好推荐效果的同时, 其实现和维护代价都比较低, 因而得到了最广泛的应用.

例如, MovieLens系统[3,9]利用 CF做电影推荐, Amazon.com[2]利用 CF推荐书籍, CD等产品, 还有为用户推荐新闻和电影的 GroupLens[4]等等. 协同过滤又可以分为基于用户的协同过滤和基于项的协同过滤, 前者认为给定用户会喜欢那些与他们有相似喜好的用户所喜爱的项, 而后者则认为给定的项会被那些喜欢与它们相似的项的用户所喜欢[2-4].

推荐的准确性是推荐系统是否有效的一个重要评判标准, 然而在实际应用中还要综合考虑算法的时空复杂性, 随着项和用户数目的增加导致算法实现的可扩展性等等问题. 例如, 因为实际系统中的用户数往往比项数有更高的数量级, 所以基于用户的协同过滤算法会面临扩展瓶颈. 而在一个项数目更高的系统中, 如果使用基于项的协同过滤算法也会面临同样的问题. 更多的情况下, 同一个系统中项的数目与用户数目是时高时低的, 这就需要一种可以随时应变的协同过滤算法, 当项数目过高时就采用基于用户的协同过滤, 而当用户数目过高时就采用基于项的协同过滤.

本文提出一类基于二部图投影和排序的协同过滤算法框架(Bipartite Graph Projection and Ranking, BGPR)并提出基于此框架的一个具体实现. 在 BGPR中, 首先用二部图表示用户对项的喜好关系, 二部图的两类顶点分别表示用户和项, 通过对二部图进行一维投影就可以将其转化成一个表示用户间或项之间相似关系的图. 再根据特定的排序算法对投影图中的节点进行排序. 不用预测用户对项的可能打分, 就可以产生推荐序列. 而且, BGPR算法既可以实现基于项的协同过滤又可以实现基于用户的协同过滤, 即分别将项作为投影图中的节点为给定用户对所有项进行排序的方法和将用户作为投影图中节点为给定项进行用户排序的方法. 本文主要贡献在于:

提出基于 BGPR的协同过滤算法框架, 因为可以采用不同的二部图投影方案和对节点的排序方法, 所以基于此框架有可以有许多不同的协同过滤算法实现.

收稿日期: 2009-02-18 基金项目: 国家自然科学基金项目(60775037)资助; 国家"八六三"高技术研究发展计划项目(2009AA01Z132)资助. 作者简介: 刘 淇, 男, 1986年生, 硕士研究生, 研究方向为数据挖掘、推荐系统、社会网络; 陈恩红, 男, 1968年生, 博士, 教授, 博士生导师, 研究方向为语义 Web, 机器学习与数据挖掘、网络信息处理、约束满足问题.

提出一个具体的 BGPR算法实现,其中利用结构相似作为二部图投影时的连边权重学习方法,然后将待推荐对象在投影图上进行带重启动的随机游走(RWR, random walk with restart),根据待推荐对象停留在每个节点的概率对图中的节点进行排序.

通过在一个推荐系统的通用数据集 MovieLens[9](由 Minnesota大学的 GroupLens研究小组提供)上的实验结果验证了 BGPR算法比其它优秀算法有更高的推荐准确度.

2 相关工作

近十年来,已经有大量基于协同过滤的推荐系统出现在科研或实际应用中.一部分学者提出了一种启发式的、根据用户以往的所有评价去推荐的 Memory-based方法.这种方法的一般思路是先为每个用户寻找相似的用户,再根据相似用户对给定项的打分和用户相似度进行项的排序[1],然而, Memory-based方法不能很好地应对数据稀疏的情况而且其可扩展性往往比较差.而另有一部分学者研究了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的 Model-based方法,例如到目前为止已经有人提出了基于贝叶斯网络的方法[11],基于最大熵的方法[10]等等,但是 Model-based方法经常需要花费大量时间建立和更新模型,而且其模型经常不能像 Memory-based方法一样覆盖所有的用户和项.

近来,也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[15,17-19].在其它领域中,[12]总结和比较了大量的计算图中结点相似的方法用来进行连边预测.[13]也提出了一种新的度量网络中顶点结构相似的方法.

在图结构中,经常用随机游走(Random walk)对节点间的相似度进行衡量[15,16,19],该方法以两个节点间的平均首达时间(ACT, average commute time)为标准.但 ACT的问题在于它对图中远离节点 i的部分有很强的依赖,即使节点是紧密相连的时候也是一样,所以经常会造成计算得到的相似度与实际相似度有较大偏差.[22]利用每次游走有限步作为抵消该依赖的一种方法.另一种抵消方法是,让从节点 i到节点 i的游走周期性"重启动",即每一步以一定的概率 c返回 i重新走步,这样几乎不会走到图中偏远的部分.随机重启也是进行网页排序的 PageRank算法[14]的基本思想.[18]用带重启动的随机游走(RWR, random walk with restart)来发现已知多媒体对象中各个媒体属性间的关联,而[21]则用 RWR来发现情感与音乐属性的关联从而进行音乐推荐.类似地,[17]也将 PageRank算法的思想用到推荐系统中.

实际应用中,推荐算法不仅要求较高的预测准确度,还要具备良好的扩展性,但是目前的推荐算法都有其不足之处.例如,基于聚类的方法往往不能有效地覆盖所有的项或者用户,而基于图的方法虽然可以为所有的用户-项对产生相似排序,但往往消耗大量的空间资源,可扩展性较差,而且其推荐准确度也有待提高[15,19].

基于上述分析,本文提出了基于二部图投影和对投影后得到的关联图中节点进行排序的 BGPR推荐算法.下面,通过基于结构相似进行二部图投影和 RWR进行节点排序的一个具体实现,详细介绍 BGPR算法框架.

3 BGPR算法描述

协同过滤是为给定用户推荐其可能感兴趣的项,但也可以将给定的项推荐给可能对其感兴趣的用户.对于前者, BGPR将投影得到由项组成的关联图,然后用给定用户的用户向量在相应的项关联矩阵上进行 RWR,从而得到所有项的排序,排名靠前即得分较高的项将被推荐给用户(Item-Rank,因为实验中使用的 item为电影,所以又称 Movie-Rank).而对于后者, BGPR将投影得到用户组成的关联图,然后用给定项的项向量在相应的用户关联矩阵上进行 RWR,得到所有用户的排序,该项将被推荐给得分较高的用户(User-Rank).本文, I表示项的集合, U表示用户的集合; n, m分别表示项和用户的个数.

3.1 二部图投影

将用户与项之间的关系表示为一个二部图 $G=(X, E)$,其中顶点集 $X=U\cup I$,用户 U_p如果对项 I_q表现出兴趣,则为 U_p与 I_q建立一条连边.然后,将该二部图分别在两个维度(用户,项)上进行投影,得到对应的一维投影图.投影后节点 i与 j间的连边权重 $\sigma(i,j)$表示节点 i与 j的相似度.用矩阵 CC表示投影图对应的关联矩阵, $CC_{ij}=\sigma(i,j)$.投影的过程会造成原二部图信息的损失,因此如何计算 $\sigma(i,j)$使其能最好地揭示出节点 i, j在原二部图中的相似情况不仅是投影过程中的关键也是对节点正确排序的重要保证.本文选择利用结构相似的计算方法来求解 σ以及 CC,实验中选择下面四种结构相似计算方法:

$$\sigma_{\text{uniform}}(i,j)=|\Gamma_i\cap\Gamma_j| \tag{1}$$

$$\sigma_{\text{jaccard}}(i,j)=\frac{|\Gamma_i\cap\Gamma_j|}{|\Gamma_i\cup\Gamma_j|} \tag{2}$$

$$\sigma_{\text{cosine}}(i,j)=\frac{|\Gamma_i\cap\Gamma_j|}{\sqrt{|\Gamma_i|*|\Gamma_j|}} \tag{3}$$

$$\sigma_{\text{min}}(i,j)=\frac{|\Gamma_i\cap\Gamma_j|}{\min(|\Gamma_i|,|\Gamma_j|)} \tag{4}$$

其中, Γ_i为投影前节点 i的邻居节点集合. $\sigma(i,j)$可以分别由公式(1)(2)(3)(4)得到.图 1(b), 1(c)就是二部图 1(a)根据公式(1)得到的两个投影.

这里的每个投影图就是项或者用户组成的关联图.相应地,可以得到关联图对应的矩阵 CC,其中当 $i\neq j$时, $CC_{ij}=\sigma(i,j)$,且 $CC_{ii}=0$.为了与 RWR相结合,再对 CC以列为单位进行归一化得到随机矩阵 C, C可以视为关联图的关联矩阵, C_{ij}表示节点 j对于节点 i的关联系数,即对节点 i而言节点 i的重要程度.这样,关联图中所有节点对间的关联程度都可以用 C中相应的元素表示.可以看到 CC是一个对称矩阵,而 C则不再具有这个性质.

例如,图 1(b)对应的用户关联矩阵 C就可以表示为:

$$C_U=\begin{bmatrix} 0 & 0.200 & 0.273 & 0.222 & 0.167 & 0.143\\ 0.125 & 0 & 0.091 & 0.111 & 0.167 & 0.143\\ 0.375 & 0.200 & 0 & 0.333 & 0.333 & 0.286\\ 0.250 & 0.200 & 0.273 & 0 & 0.167 & 0.286\\ 0.125 & 0.200 & 0.182 & 0.111 & 0 & 0.143\\ 0.125 & 0.200 & 0.182 & 0.222 & 0.167 & 0 \end{bmatrix}$$

3.2 排序算法

BGPR算法的目标是通过排序的方式估计用户与项之间的关系. BGPR用向量表示查询节点,向量中的每一维都代表关联图中的一个节点,其值则表示相应节点与该查询节点的亲和度. BGPR采用带重启动的随机游走(RWR)来预测一个节点(或向量)Y对查询节点(或向量)X的重要性或关联程度.考虑从节点向量 X开始的随机游走,每走一步保证两件事情:首先,如果一个节点 Z与节点 P联系紧密,而 X又是与 P相关联的,则通过 P, X与 Z也会建立一定的联系.其次,由于 X是通过间接的方式与 Z建立联系的,所以相比较 P与 Z, X与 P, X与 Z的联系强度会有一定的"衰减".

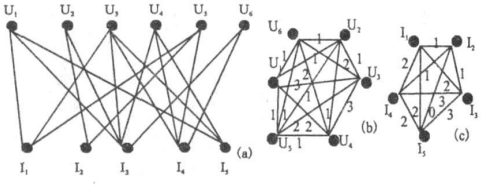

图1　用户与项组成的二部图(a),以及它在用户维与项维的两个投影(b)(c)

Fig.1　A bipartite graph about users and items (a), with its two projections on users (b), items (c)

或者说, X游走到 P以后会以一定的概率沿 P走向与 P有联系的节点.而除此之外,它还有可能返回节点 X重新走步,来消除远端依赖[12].定义 steady-state概率 $S_X(Y)$表示从节点 X开始的随机游走到达节点 Y的概率.那么 $S_X(Y)$就表示了 Y相对于 X的亲密程度.

定义 O_q为查询对象,它可以是一个用户或一个项,如果 O_q是一个用户,BGPR算法要得到与它关系最紧密的项,而如果 O_q是项,BGPR算法得到与它关系最紧密的用户.这里若关联图或关联矩阵 C由 k个节点组成,则 O_q对应的 steady-state概率向量 $S_q=(S_q(1),\cdots,S_q(k))$即为所求.定义归一化向量 V_{O_q}是根据 O_q在训练集中的记录而建立的节点向量,在归一化 V_{O_q}之前向量 V'_{O_q}第 j维的元素对应于:

$$V'_{O_q}{}^{j}=\begin{cases}1 & \text{if } O_q \text{ is user and } O_q \text{ rated } I_j \text{ or}\\ & \text{if } O_q \text{ is item and } U_j \text{ rated } O_q\\ 0 & \text{otherwise}\end{cases} \quad (5)$$

由3.1和3.2本文设计的推荐算法 BGPR如图2所示.

在推荐算法 BGPR中,可调参数 c为随机游走重启动的概率,(1-c)为衰减因子,用来衡量信息的衰减程度.实际应用中,平均情况下, t=20时, S_{O_q}即可收敛[17].

考虑一个简单的例子,如果根据结构相似公式(1)进行二部图投影然后对图1(a)中的用户 U_6做推荐预测. U_6的节点向量 $V_{u_6}=(0,0,0.5,0.5,0)$,运行 BGPR算法,得到一个关于项的 steady-state概率向量 $S_{U_6}=(0.002357, 0.001176, 0.496886, 0.496674, 0.002903)$.此时 I_5最有可能推荐给 U_6.作为验证,再对 I_5做用户推荐预测,则其节点向量 $V_{I_5}=(0.333,0,0.333,0.333,0,0)$,再次运行 BGPR算法,得到一个关于用户的 steady-state概率向量 $S_{I_5}=(0.331319, 0.001089, 0.332028, 0.331414, 0.001391, 0.001755)$,与前边的结果相符,此时可以发现 I_5最有可能被推荐给 U_6.

输入:表示用户与项之间关系的二部图 G,待推荐对象 O_q,参数 c
输出:对 O_q的推荐序列

1. 根据公式(1)(2)(3)或(4)得到 G的一维投影图对应的关联图,若 $O_q\in U$,则投影得到关于项的关联图,若 $O_q\in I$,则投影得到关于用户的关联图
2. 令关联图对应的矩阵 CC对角线的元素为0,再以列为单位将其归一化,得到关联矩阵 C
3. 由(5)得到向量 V'_{O_q},并对其归一化得到向量 V_{O_q}
4. t=0,初始化 $S_{O_q}=V_{O_q}$
5. while(S_{O_q}不收敛 || t<MAX_LOOP)
6. {
7. $S_{O_q}=(1-c)*C*S_{O_q}+c*V_{O_q}$
8. t++
9. }
10. 按照大小将 S_{O_q}中的元素排序
11. 若 $S_{O_q}(j)$排名靠前且 $V_{O_q}(j)=0$,则将 $S_{O_q}(j)$对应的项(用户)推荐给 O_q,得到 O_q的推荐序列

图2　推荐算法 BGPR流程

Fig.2　Flow chart of BGPR algorithm

4 实　验

MovieLens[3,9]是一个进行电影推荐的网站, GroupLens小组从中整理出了适用于各种推荐场景的一个标准数据集.在本文使用的数据集中,对用户进行了处理,只考虑那些对超过20部电影打分的用户,总共包含943个用户(m=943)对1 682部电影(n=1 682)的评分,共约100 000个评分.

实验中使用数据集的方法与[15, 17, 19]中类似:对于参数训练部分(主要是对参数 c的测试),将整个数据集分成训练集与测试集两部分,而测试部分则使用五对数据集进行5-交叉测试,每次用80%的评分数据做训练集,20%的评分数据作为测试集.

4.1 评价标准

本文采用 DOA(degree of agreement)[20]为标准对实验结果进行评价.根据使用对象的不同将其扩展为用户的 DOA(U-DOA)和项的 DOA(I-DOA),其思想是:

对于 U-DOA,先定义 $NW_{u_i}=n/(L_{u_i}\cup T_{u_i})$,其中 n为所有项的个数, L_{u_i}为用户 U_i在训练集中评价过的项集合, T_{u_i}为 U_i在测试集中评价的项集合.

$$check\_order_{U_i}(I_j,I_k)=\begin{cases}1, & \text{if}(predict\_rank_{I_j}\geqslant predict\_rank_{I_k})\\ 0 & \text{otherwise}\end{cases}$$

从而对于用户 U_i其 DOA得分定义如下:

$$DOA_{U_i}=\frac{\sum_{(I_j,I_k)\in NW_{U_i}} check\_order(I_j,I_k)}{|T_{U_i}|*|NW_{U_i}|} \quad (6)$$

I-DOA的定义与 U-DOA类似,这里不再赘述.通过定义可以看出,随机预测的 DOA值大约为50%,而理想情况下

的 DOA值为 100%. 并且, 实验中以所有用户 (项)的 DOA取平均作为总体的效果评价.

$$U\text{—}DOA=\frac{\sum_{U_i} DOA_{U_i}}{|U|} \quad 或 \quad I\text{—}DOA=\frac{\sum_{I_i} DOA_{I_i}}{|I|}$$

4.2 参数选择及实验结果

在 PageRank算法中重启动概率 c经常取为 0.15, 而在实验中, 通过各个方法对 c在 (0,1)之间变化时的表现情况可以发现, c越接近于 1, BGPR算法的推荐效果越好. 其实这一方面与所用的关联图的半径有关[18], 一方面与数据的性质有关, 在当前的 MovieLens数据集中, 项 (用户)的相似度绝大部分比例依赖于项 (用户)间的直接路径数. 图 3显示了随着 c的变化, 几种二部图投影方法对应的 BGPR算法的表现. 接下来的实验中若无特殊说明, 选择 c=0.99.

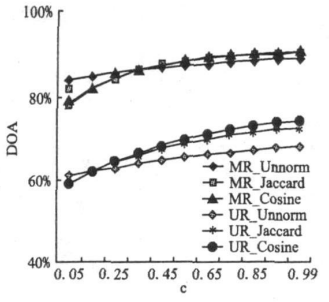

图 3 随着 c的变化各个方法的表现

Fig. 3 BGPR performance under different c

表 1 Movie—Rank 5交叉测试结果

Table 1 Movie—Rank performance on 5-fold cross validation

	METHOD	SPLIT1	SPLIT2	SPLIT3	SPLIT4	SPLIT5	MEAN
Movie—Rank DOA (%)	Unnorm	89.05	89.12	89.29	89.09	88.88	89.09
	Jaccard	90.29	90.37	90.64	90.38	90.17	90.37
	Cosine	90.57	90.68	91.02	90.67	90.53	90.69
	Min	80.98	80.40	81.65	80.34	80.16	80.71

表 1、表 2显示了不同的结构相似对应的 Movie—Rank和 User—Rank方法在进行 5交叉测试时的表现.

表 3显示了使用 MovieLens作为数据集的不同推荐算法的效果比较, 这些算法的具体细节和实现可以参见[15,17,19]. 这里 BGPR算法选择基于 Movie—Rank的 Cosine结构相似 (MR—Cosine)为代表与其他算法进行比较, 其中的 ItemRank[17]与本文基于公式 (1)的 Movie—Rank实现方法类似.

表 2 User—Rank 5交叉测试结果

Table 2 User—Rank performance on 5-fold cross validation

	METHOD	SPLIT1	SPLIT2	SPLIT3	SPLIT4	SPLIT5	MEAN
User—Rank DOA (%)	Unnorm	75.52	81.15	80.99	79.94	76.40	78.80
	Jaccard	74.06	80.75	81.11	80.56	76.86	78.67
	Cosine	74.35	81.04	81.60	80.94	77.34	79.05
	Min	67.94	75.56	76.91	75.47	72.42	73.66

对于其中的每个算法, 给出了其 5交叉测试后的平均实验结果, 以及与传统的 MaxF算法的表现差异比较. MaxF算法简单地将项按被浏览次数进行降序排序, 每次都将其没有评价过且排序最靠前的项推荐给给定用户[17]. MaxF算法也是比较的基准算法.

4.3 讨 论

首先, 由表 1、2可以发现在同一个数据集中, 将二部图投影得到电影之间的关联图然后对电影排序 (Movie—Rank)的推荐效果明显优于建立用户之间的关联图然后对用户排序 (User—Rank)的结果, 其实这与 MovieLens数据集的性质有关. MovieLens数据集对用户进行了处理只保留了至少评价了 20部电影的用户, 以保证每个用户的兴趣都可以得到确切的反映, 而数据集中的电影却没有经过相似处理. 对用户的处理可以让一般的推荐算法防止 "冷启动"问题但是 BGPR算法采用了 RWR做为距离评判标准, 可以有效地防止 "冷启动"问题. 而且对用户的预处理使得 BGPR构造的用户关联图过于稠密 (数据显示用户关联图对应的用户关联矩阵中非零元素的比例在[0.922, 0.934]区间, 而此比例在项关联图中仅为[0.625, 0.642]), 从而降低了用户与用户间的区分度, 造成了对用户排序 (User—Rank)的不准确. 图 4显示了电影度数的幂律分布情况, 图中电影度数服从 $\lambda=0.8$的幂律分布 ($P(k)\sim k^{\lambda}$), 这说明 MovieLens数据集的电影网络可以反映实际情况中的电影度数分布, 同时根据这种实际数据所得到的电影关联图可以有效地反映电影间的相似情况, 即 Movie—Rank算法的优秀推荐效果在实际应用中也可以得到. 总之, 利用 RWR可以有效地防止 "冷启动"问题, 而且根据实际数据构造的关联图可以与 RWR紧密融合, 使得 BGPR算法有巨大的实用价值.

表 3 不同算法效果比较

Table 3 Comparison between different algorithms

	Algorithm	DOA (%)	Difference with MaxF (%)
Movie—Rank	MAXF	84.07	0
	CT	84.09	+0.02
	PCA CT	84.04	-0.03
	One-way	84.08	+0.01
	Return	72.63	-11.44
	L^+	87.23	+3.16
	ItemRank	87.76	+3.69
	Katz	85.83	+1.76
	BGPR(MR—Cosine)	90.69	+6.62

在计算效率方面, 给定关联矩阵 C时, BGPR只要经过较少次数的迭代就可在 $O(n^2)$或 $O(m^2)$时间里得到一次推荐, 而 L^+或者 CT算法只能一次得到所有推荐, 所以只能定时更新推荐而无法做到实时更新推荐. 但 BGPR算法的一个问题在于, 如果关联矩阵 C中有一个节点发生了更新, 则整个关联矩阵都需要更新, 这将消耗 $O(n^2)$或 $O(m^2)$的时间, 所以可以选择定时更新关联矩阵 C而实时更新节点向量的方法, 以得到更好的推荐效果.

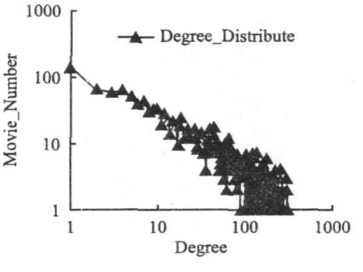

图 4 电影度数的幂律分布

Fig. 4 Power-law distribution of Movies' degree

最后是算法的空间复杂度和可扩展性.当基于项过滤对项进行排序时,关联矩阵 C的大小与用户数无关,这是一个很有用的性质.因为一般的应用中,用户数目比项的数目有更高的数量级[17],而且一般用户可能只对少数的一些项感兴趣,可以将用户的兴趣直接用链表的形式表现出来,从而节省巨大的空间资源.如果用 k_u表示每个用户平均感兴趣的项个数,那么此时算法的空间复杂度为 $O(n^2+m*k_u)$,即使用户数目不断增加,用户对项的兴趣也不断增加,BGPR算法都可以有效地应用.当有新的项时,BGPR和 ItemRank只需要 $O(n)$的空间消耗,其 L^+与 CT等则需要 $O(m+n)$.而当增加一个新用户时 ItemRank和 BGPR需要 $O(k_u)$的空间,而 L^+,CT等算法却需要 $O(m+n)$的空间,问题在于在一般的应用中 $m\gg n\gg k_u$,而基于用户过滤的用户排序,虽然空间消耗可能要高于对项排序的方法,但仍然小于 L^+与 CT等方法,这里不再赘述.

5 结 论

本文展示了一种基于二部图投影与排序相结合的协同过滤推荐算法 BGPR,它可以预测每个用户可能喜欢的项,也可以将项推荐给那些可能喜欢它的用户.更重要的是 BGPR是一个协同过滤的框架算法,基于不同的二部图投影和排序方案可以产生不同的推荐算法.文中采用结构相似进行二部图投影再利用带重启动的随机游走(RWR)对图中的节点进行排序从而产生推荐序列,通过带重启动的随机游走可以使得投影图的节点相似信息得到扩散从而防止"冷启动"问题而且提高了推荐准确度.通过在同一个数据集上与其它的一些优秀算法的比较证实了 BGPR不仅有更高的推荐准确度,同时保持较低的时空复杂度,较好的可扩展性,而且可以适用于不同情况的针对海量数据的实际推荐应用中.

由结构相似得到的投影图可以用来描述原二部图所隐含的节点间的信息.但是这类相似计算方法是对称性的,认为 $\sigma(i,j)$等于 $\sigma(j,i)$,即节点 i视对方是同等重要的.其次,结构相似简单地视节点 i的所有邻居节点对 i的相似程度有相同的影响,然而,假设节点 $s\in\Gamma_i\cap\Gamma_j$且 $\Gamma_i\gg\Gamma_j$,则显然邻居节点 s应当比邻居结点 s更能说明节点 i相似.由于结构相似存在着以上种种缺点和不足,使其不能完全地反映原二部图所隐含的节点间的相似信息.为了更准确地挖掘原二部图的信息,寻找新的二部图投影方案将是未来工作的主要努力方向.

References:

[1] Gediminas Adomavicius, Alexander Tuzhilin. 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, June 2005, 17(6): 734-749.

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

[3] Miller B N, Albert I, Lam S K, et al. MovieLens unplugged: experiences with an occasionally connected recommender system[C]. Proceeding of Int'l Conf. Intelligent User Interfaces, Miami, USA, 2003. 263-266.

[4] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]. Proceeding of the CSCW Conference, Chapel Hill, NC, 1994. 175-186.

[5] 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.

[6] Balabanovic M, Shoham Y. Fab: content-based, collaborative recommendation[J]. Comm. ACM, 1997, 40(3): 66-72.

[7] Pazzani M. A framework for collaborative, content-based and demographic filtering[J]. Artificial Intelligence Rev., Dec. 1999, 13: 393-408.

[8] Souvik Debnath, Niloy Ganguly, Pabitra Mitra. Feature weighting in content based recommendation system using social network analysis[C]. Proceedings of WWW, Beijing, China, April, 2008, pages 1041-1042.

[9] http://www.movielens.umn.edu.

[10] Pavlov D, Pennock D. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains[C]. Proceedings of 16th Ann. Conf. Neural Information Processing Systems (NIPS'02), Canada, 2002. 1441-1448.

[11] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, 1998. 43-52.

[12] David Liben-Nowell, Jon Kleinberg. The link-prediction problem for social networks[J]. Journal of the American Society for Information Science and Technology, 2007, 58: 1019-1031.

[13] Leicht E A, Petter Holme, Newman M E J. Vertex similarity in networks[J]. Physical Review E, 2006, 73: 026120.

[14] 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.

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

[16] Luh Yen, Marco Saerens, Amin Mantrach, et al. A family of dissimilarity measures between nodes generalizing both shortest-path and the commute-time distances[C]. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Las Vegas, USA, 2008. 785-793.

[17] Marco Gori, Augusto Pucci. A random-walk based scoring algorithm with application to recommender systems for large-scale e-commerce[C]. Proceedings of WEBKDD'06, Philadelphia, USA, 2006. 127-146.

[18] Pan J Y, Yang H J, Faloutsos C, et al. Automatic multimedia cross-modal correlation discovery[C]. Proceedings of ACM Int'l Conference on Knowledge Discovery and Data Mining (KDD'04), Seattle, USA, 2004. 653-658.

[19] Fouss F, Pirotte A, Renders J M, et al. A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering[C]. Proceedings of IEEE/WIC/ACM International Joint Conference on Web Intelligence, Compiegne, France, 2005. 550-556.

[20] Siegel S, Castellan J. Nonparametric statistics for the behavioral sciences[M]. New York, USA: McGraw-Hill College, 1988.

[21] Kuo F F, Chiang M F, Shan M K, et al. Emotion-based music recommendation by association discovery from film music[C]. Proceedings of the 13th Annual ACM International Conference on Multimedia, Singapore, 2005. 507-510.

[22] Sarkar P, Moore A. A tractable approach to finding closest truncated-commute-time neighbors in large graphs[C]. Proceedings of UAI, Vancouver, BC, Canada, 2007.