

融合用户时效偏好的推荐算法

李 贵 陈盛红 李征宇 韩子阳 孙 平
(沈阳建筑大学信息与控制工程学院 沈阳 110000)

摘 要 在推荐系统中,随时间精确捕获用户偏好能有效提高推荐精度。但基于所有用户的简单时间相关性通常是没有实际意义的,因为不同用户的偏好随着外部环境不同而发生改变。用户时下偏好受用户长期偏好和短期偏好的共同影响。为了捕获用户长期和短期偏好,在推荐系统中引入基于会话的时态图 STG(Session-based Temporal Graph),提出基于 STG 的路径融合算法 PFA(Path Fusion Algorithm),并生成对某个用户的 Top-N 物品推荐。使用 CiteULike 和 Delicious 两个历史数据集来评估算法的有效性,实验结果表明所提算法在准确度上要高于以往传统算法。

关键词 推荐算法,会话,时态图,偏好融合
中图法分类号 TP301.6 **文献标识码** A

Recommendation Algorithm with User Temporal Preference Fusion

LI Gui CHEN Sheng-hong LI Zheng-yu HAN Zi-yang SUN Ping
(Faculty of Information & Control Engineering, Shenyang Jianzhu University, Shenyang 110000, China)

Abstract Accurately capturing user preferences over time can improve the precision in recommender systems. Simple correlation over time is typically not meaningful, since users change their preferences due to different external events. User behavior can often be determined by individual's long-term and short-term preferences. In order to capture it, we propose the Session-based Temporal Graph(STG). Based on the STG model framework, we propose the Path Fusion Algorithm(PFA) to make accurate top-N recommendation. And we evaluate the effectiveness of our method using two real datasets, CiteULike and Delicious. Experimental results show that the proposed algorithm has higher accuracy than most of previous algorithms.

Keywords Recommendation algorithm, Session, Temporal graphs, Preference fusion

1 引言

推荐系统的时效性在现实世界应用中是非常重要的,例如京东商城等电子商务网站的产品推荐,Netflix 等视频网站的视频推荐,谷歌和新浪等媒体网站的新闻推荐。但关键问题是如何随时间精确捕获用户偏好?

用户的偏好具有多样性和实效性等特性,这给个性化时效推荐系统带来了极大的挑战。用户的整体行为可能取决于他的长期兴趣,但是在任意给定的时间,用户同样受瞬时事件产生的短期兴趣影响。例如新产品的发布和过生日等特殊的个人场合等。所以,一个好的时效推荐系统需要综合用户的长期和短期偏好。

时间维度是挖掘时效信息的一个重要途径。很多时变模型把时间作为所有用户时效信息的一个维度。然而,本文认为时间维度只是一个局部的影响,不同用户具有不同时间维度。例如,在同一时间段、不同外部环境下,用户购买不同的物品,那么这些物品与所有用户在时间上的相关性就没有意义,因为这些事件是完全不同的。但在另一方面,可能在一段较短时间内,用户在相同外部环境下购买多样的物品。这样,

这些物品在时间上的相关性对于某些用户而言也是非常重要的。

很多推荐系统只处理用户显性反馈(如评分),但显性反馈并不总是能够获取的,并且在很多的实际情形下,系统是没有显性反馈的。那么推荐系统必须处理隐性反馈,包括用户购买记录、浏览记录等。本文专注为用户长期和短期偏好的隐性数据集进行显性建模。其中面临的两个问题是:

- 1) 如何描述用户的长期偏好和短期偏好?
- 2) 如何建模两种偏好的相关度?

针对这两个问题,本文提出了基于会话的时态图(STG),并应用路径融合算法 PFA 来计算 STG 中的 Top-N 推荐列表。模型的特点是引入了会话节点,用来表示用户购买物品的时间信息。也就是说,STG 通过用户和物品的关联计算用户长期偏好,通过用户和会话的交互记录关联来计算用户短期偏好^[6]。

本文的主要工作包括:

- 1) 引入一个基于会话的时态图(STG),将用户长期和短期偏好基于时间信息进行建模。
- 2) 在 STG 框架基础上,提出了一种路径融合算法

本文受国家自然科学基金(61070024)资助。

李 贵(1964—),男,博士,教授,主要研究方向为 Web 数据挖掘与信息集成、分布对象技术、软件工程, E-mail: Ligui21c@sina.com; 陈盛红(1991—),男,硕士生,主要研究方向为 Web 数据挖掘和推荐系统。

(PFA),用于平衡用户长期和短期偏好对准确推荐的影响。

3)文中通过两个历史数据集比较本文所提出的算法和协同过滤算法。结果证实,利用STG将时效数据与图相结合提出的路径融合算法(PFA),在推荐系统中可平衡用户长期和短期偏好,有效提高推荐的准确度。

文中使用主要符号如表1所列。

表1 主要符号意义

符号	描述
G	三分图SGT
E	G的边集
U,I,S	用户节点集,物品节点集,会话节点集
v_u, v_i, v_{it}	用户节点,物品节点,会话节点
ω	边的加权函数
$N(u)$	用户 u 查看的所有物品
$N(u,t)$	用户 u 在时间 t 查看的所有物品
P_i	用户节点到一未知物品节点在SGT中的路径
P_{it}	用户节点到一未知会话节点在SGT中的路径
$P(u,i)$	v_u 到 v_i 的路径集
$P(u,s)$	v_u 到 v_{it} 的路径集
$\Gamma(P)$	路径P的权重
$\varphi(v_k, v_{k+1})$	v_k 到 v_{k+1} 的传递函数
$out(v)$	节点 v 的出度
ρ	用来控制节点出度在偏好传递中影响的参数
β	调整物品和会话节点间路径融合比例
η	调整用户节点到物品/会话节点的边缘权重参数

2 相关工作

本节主要介绍时效推荐算法的相关研究工作,并简单描述基于图的推荐算法。

2.1 时效推荐

在将时间信息融入推荐算法中,文献[1]计算所有物品的时间权重,并根据用户短期行为生成推荐结果。但它忽略用户长期行为在推荐系统中的作用。在某个时间段内,用户短期行为并不能完全决定用户时下偏好,它还受用户长期行为影响。文献[1]利用规范化的协同过滤(CF)算法处理一个随时间迭代的预测问题,并在Netflix实验中发现提高预测准确度的算法,当训练数据集和测试数据集增加时,准确度却没有得到同样提高。与其它算法相比,本文所提出的算法根据问题域大小设置不同的全局参数,使算法能够适应不同外部环境。本文通过不同方式处理用户时间信息,对用户长期偏好和短期偏好进行建模来平衡两种偏好对推荐的影响。

文献[5]提出基于“修正”的推荐方法,该方法通过挖掘用户历史记录和用户明确定义的一组稳定兴趣偏好来获取用户长期偏好,并通过引入“修正”的方式让用户明确表达短期偏好。文献[3]在Netflix中提出了一个基于时态的矩阵分解模型来预测用户的视频评分预测。本文通过图形建模用户长期和短期偏好,并在隐性反馈数据集基础上生成Top-N推荐列表。

2.2 基于图的推荐

本文将三分图引入推荐系统中,以图的形式去建模用户、时间和物品之间的交互关系,并提出计算节点间的相关性的算法,与用户 u 相关性越高的物品在用户 u 的推荐列表中权重越大。

度量图中两个节点之间相关性的方法很多。文献[7]建立了一个视频合作视图,在半监督学习下通过标签传播方法在YouTube上进行视频推荐。文献[8]提出一种基于二分图的模型来计算用户和物品之间关联强度,并应用到物品推荐中。文献[9]提出一种在图表上结合时态数据进行高级检索的方法,将时间分成若干段,然后在时间段上建立与前后相邻

两个时间段相邻的表格。一般来说图中节点的相关性主要取决于3个因素:1)两节点之间的路径数;2)两个节点之间路径的长度;3)两个节点之间的路径经过的节点数。那么用户 u 对物品 i 有较高偏好度需满足的条件:1) u 和 i 之间的路径很短;2) u 和 i 之间存在很多路径,并且这些路径都不太长;3) u 和 i 有避开高入度和高出度节点的路径。PFA是以用户_物品_时间三分图为基础的,采用的是局部搜索策略[2],本文将依据这些规则来设计PFA算法。

3 STG建模

本节主要介绍如何使用3种不同的节点类型和不同权重的边将用户长期和短期偏好在图中建模。

数据类型是(用户,物品,时间)三元组,通常使用三维张量模型表示。然而,在推荐系统中,时间维度却是局部的,只需将同一时间段或相近时间段内用户进行交叉比较,而不是所有用户。因为在同一时间段或相近时间段内,两用户对同一物品或相似物品产生了行为,这更能体现两用户偏好相似。

本文建立一个新的节点类型——会话节点(物品和特定时间段相结合),来表示用户和会话的联系。在模型中,将时间划分若干段,并将物品与相应的时间段绑定,将(用户,物品,时间)转变成(用户,物品)和(用户,会话)。需要注意的是,会话的长度可以是一个小时、一天或一星期等等,这取决于算法所面临的问题域大小。

在(用户,物品)和(用户,会话)基础上,提出基于会话的时态图(SGT)。SGT是个有向三分图 $G(U,S,I,E,\omega)$,这里 U 表示一组用户节点, S 表示一组会话节点, I 表示一组物品节点, E 表示 G 的边集, ω 表示边的非负加权函数。图1是一个包含4个用户节点、3个会话节点和2个物品节点的SGT简单例子。它表示物品 i_1 分别与用户 u_1, u_2 和 u_3 关联,物品 i_2 分别与用户 u_3 和 u_4 关联。此外,用户 u_1 和 u_2 与会话 s_1 关联,因为用户 u_1 和 u_2 在时间 t_1 同时查看物品 i_1 ;用户 u_3 和 u_4 与会话 s_4 关联,因为用户 u_3 和 u_4 在时间 t_2 同时查看物品 i_2 。

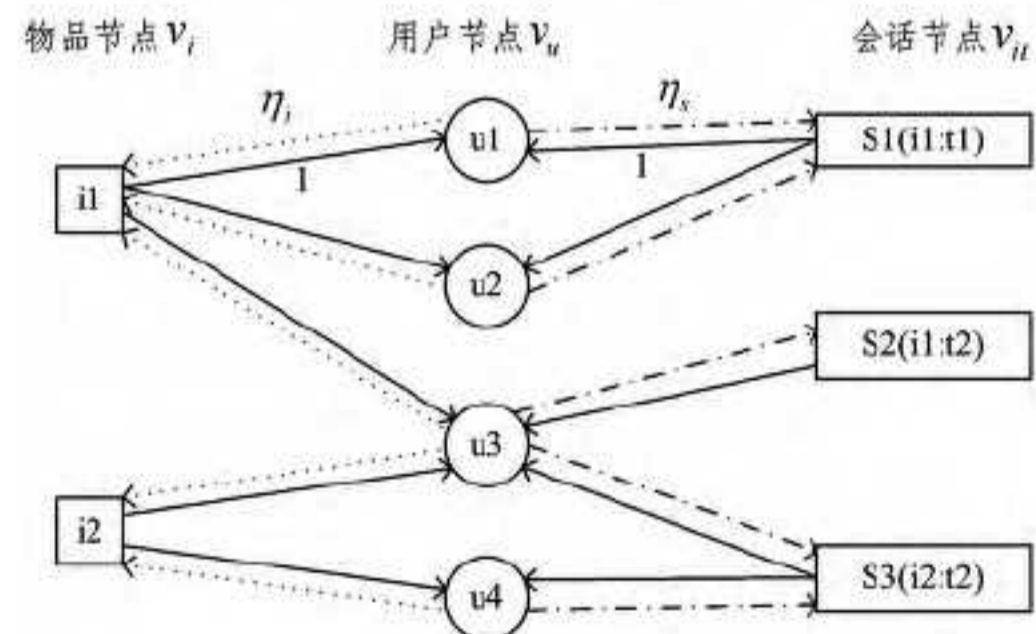


图1 STG实例

在STG中,与用户节点 v_u 关联的所有物品节点 v_i 定义为 $N(u)$,表示用户 u 的长期偏好,与用户节点 v_u 关联的所有会话节点 v_s 的物品定义为 $N(u,t)$,表示用户 u 在时间 t 的短期偏好。因此,从用户节点 v_u 沿着两方向,经过 $N(u)$ 或 $N(u,t)$,然后到达未知物品节点或会话节点。这样,到达未知物品节点代表用户的长期偏好,到达未知会话节点代表短期偏好。

STG中边的权重定义:

$$\omega(v, v') = \begin{cases} 1, & v \in I \cup S, v' \in U \\ \eta_i, & v \in U, v' \in I \\ \eta_s, & v \in U, v' \in S \end{cases} \quad (1)$$

该定义意味着,无论从物品节点或会话节点出发的边 $e(v, v')$, 它的权重都是 1, 因为物品和会话节点只与用户节点关联。对于从用户节点出发的边, 如果它连接物品节点, 其权重是 η_i ; 如果它连接会话节点, 其权重是 η_s 。本文使用不同的权重来模拟用户长期和短期偏好对时效偏好的影响, 以达到更高推荐精度。如果两用户只通过物品节点关联, 用户_用户相似度由长期偏好计算得到, 如果两用户通过会话节点关联, 用户_用户相似度由短期偏好计算得到。

4 基于 STG 推荐

本节通过 STG 中物品节点和会话节点来建模用户长期和短期偏好, 建立两种偏好的联系, 通过路径融合算法(PFA)生成 Top-N 推荐列表。

4.1 路径融合算法(PFA)

通过用户长期和短期偏好的关联模型给用户 u 在时间 t 做推荐时, 本文基本思想是考虑用户 u 与物品节点 v_i 和会话节点 v_{it} 的联系, 通过加权融合两张推荐列表。用户通过用户节点将偏好传递到物品 $N(u)$ 或会话 $N(u, t)$ 。最后传递到物品节点, 则相当于用户长期偏好, 最后传递到会话节点, 则相当于用户短期偏好。串联 STG 中从源节点到未知物品节点或会话节点的传播路径, 用户对物品的偏好度是所有从用户节点传递到物品节点或会话节点路径加权总和, 根据偏好度大小生成 Top-N 推荐列表。

给定一个用户 u 和时间 t , 用户节点 v_u 是起始节点, 物品节点 v_i 和会话节点 v_{it} 是终止节点。注意, 会话节点 v_{it} 是从所有会话节点中选择, 因为我们采取适当的时间分段标准囊括了当前用户的大部分短期影响。如果 $P\{v_0, v_1, \dots, v_n\}$ 是源节点 v_0 到终止节点 $v_n \in \{v_i, v_{it}\}$ 的路径, 最后传递到 v_n 的偏好值定义为:

$$\Gamma(P) = \prod_{v_k \in P, 0 \leq k < n} \psi(v_k, v_{k+1}) \sigma(v_n) \quad (2)$$

式中, $\sigma(v_n)$ 是路径融合权重值, 它的值取决于终止节点类型:

$$\sigma(v_n) = \begin{cases} \beta, & v_n = v_i \\ 1 - \beta, & v_n = v_{it} \end{cases} \quad (3)$$

式中, β 用于调整长期偏好与短期偏好在最终推荐结果中所占比率。 $\beta=0$ 表示推荐结果只由用户短期偏好构成, $\beta=1$ 表示推荐结果只由用户长期偏好构成。

式(2)中 $\psi(v_k, v_{k+1})$ 是一个传递函数, 用于测试有多大偏好从 v_k 传递到 v_{k+1} 。定义如下:

$$\psi(v_k, v_{k+1}) = \left(\frac{\omega(v_k, v_{k+1})}{\sum_{v' \in out(v_k)} \omega(v_k, v')} \right)^\rho \quad (4)$$

式中, $out(v_k) = \{v' \in V : e(v_k, v') \in E\}$, 其中 $\rho \in [0, 1]$ 是用于调整出度在传递过程中的影响的参数。这个定义意味着, 如果节点 v_k 有较大的出度, 那节点 v_{k+1} 将得到较低的入度偏好。通过应用边权重定义(如式(1)), 式(4)可以精简为式(5)。

$$\psi(v, v') = \begin{cases} \frac{1}{|out(v)|^\rho} & v \in I \cup S, v' \in U \\ \left(\frac{\eta}{\eta |out(v) \cap U| + |out(v) \cap S|} \right)^\rho & v \in U, v' \in I \\ \left(\frac{1}{\eta |out(v) \cap U| + |out(v) \cap S|} \right)^\rho & v \in U, v' \in S \end{cases} \quad (5)$$

式中, $\eta = \eta_i / \eta_s$ 是一个控制偏好(来至用户节点)传递到物品

节点或者会话节点的比率的参数, 因此它在测量用户_用户相似性中影响长期偏好和短期偏好的导向。如果 $\eta \rightarrow \infty$, 两用户只能通过物品节点关联, 这意味着只有用户长期偏好影响用户_用户相似性; 如果 $\eta \rightarrow 0$, 两用户只能通过会话节点关联, 因此只有用户短期偏好影响用户_用户相似性。

STG 作为一个三分图, 用户节点到未知物品/会话节点的距离应该是等于或大于 3 的奇数。给定一个用户和一个未知物品, 在 STG 中两个节点之间会有很多路径(距离 ≥ 3)。但是本文只考虑最短路径(距离 = 3)来衡量用户对物品的偏好, 因为对最终的节点, 较长路径贡献的偏好权重小于短路径, 甚至会产生噪声, 而且短路径可以降低广度优先搜索的复杂度。

因此, 本文用 $P(u, i)$ 代表用户 u 从源节点到未知终止节点 v_i 或 v_{it} 的最短路径集(距离 = 3)。用户 u 对物品 i 的偏好度为 p_{ui} , 计算如下:

$$p_{ui} = \sum_{P \in P(u, i)} \Gamma(P) \quad (6)$$

最后, 将所有未知物品的 p_{ui} 从大到小排序, 然后返回前 N 个未知物品给用户。

PFA 是在 STG 中利用广度优先搜索算法实现的, 它的伪代码如算法 1 所示。

算法 1 PFA 的伪代码用于活跃用户在时间 t 的推荐。

数据: STG $G(U, S, I, E, w)$, 用户 u , 时间 t

结果: 在时间 t 给用户 u 推荐

Queue Q; /* 建立路径节点队列 Q */

NodeSet V; /* 节点集合 V, 包含 v_u, v_i, v_{it} */

Q.append(v_u); /* 在队列 Q 中添加节点 v_u */

distance[v_u] = 0; /* 初始化 v_u */

while Q is not empty do

Node $v = Q.top()$;

if V.contains(v) then

└ continue;

if distance[v] > 3 then /* 限定最短路径距离为 3 */

└ break;

V.insert(v); /* 添加 v */

for each $v' \in out(v)$ do

└ if !V.contains(v') then

└└ distance[v'] = distance[v] + 1;

└└ Q.append(v'); /* 遍历生成路径 */

└ if distance[v] < distance[v'] then

└└ rank[v'] = rank[v] + rank[v'] · $\psi(v, v')$;

└└ /* 计算传递偏好值 */

└ if distance[v'] = 3 then

└└ if $v' \in v_i$ then

└└└ rank[v_i] = β rank[v'];

└└ if $v' \in v_{it}$ then

└└└ rank[v_{it}] = $(1 - \beta)$ rank[v'];

└└ rank[v] = rank[v_i] + rank[v_{it}]

└└ /* 计算路径偏好值 rank[v] */

rank.sort(v); /* 排序 */

return top-N unknown items; /* 生成 Top-N 推荐列表 */

总之, 给定一用户 u , 通过三步得到从用户节点 v_u 到终止节点 v_i, v_{it} 的 4 种类型的最短路径, 如图 2 所示。

1) 用户_物品_用户_物品 (P1): 这种类型的路径从 v_u 开始, 经过 $N(u)$ (u 查看的物品) 中所有的物品, 最后通过其它用户(也查看过 $N(u)$ 的物品的用户)跳到用户 u 的未知物品集;

2) 用户_物品_用户_会话(P2): 与路径 P1 相似, 但不同的是终止节点是会话节点;

3) 用户_会话_用户_物品(P3): 这种类型的路径从用户节点 v_u 开始, 后跳过 $N(u, t)$ (用户 u 在时间 t 查看物品集合) 中所有的物品, 最后找到相似未知物品;

4) 用户_会话_用户_会话(P4): 与路径 P3 相似, 但终止节点是会话节点而不是物品节点。

总结: P1 和 P3 都终止于物品节点 v_i , 最后找到类似于 $N(u)$ 的未知物品节点, 反映的是用户 u 的长期偏好; P2 和 P4 都终止于会话节点 v_{ii} , 最后找到类似于 $N(u, t)$ 的未知物品节点, 反映的是用户 u 的短期偏好; P1 和 P2 中用户之间通过物品节点关联, 主要是根据用户长期偏好测量用户相似度; P2 和 P4 中物品之间通过会话节点关联, 主要是根据用户短期偏好测量用户相似度。

对于没有时态数据(没有会话节点)的特殊情况, 只有 P1 可应用到用户_物品二分图上。PFA 也可以在仅有物品节点导出偏好的情况下, 用于没有时态的推荐。

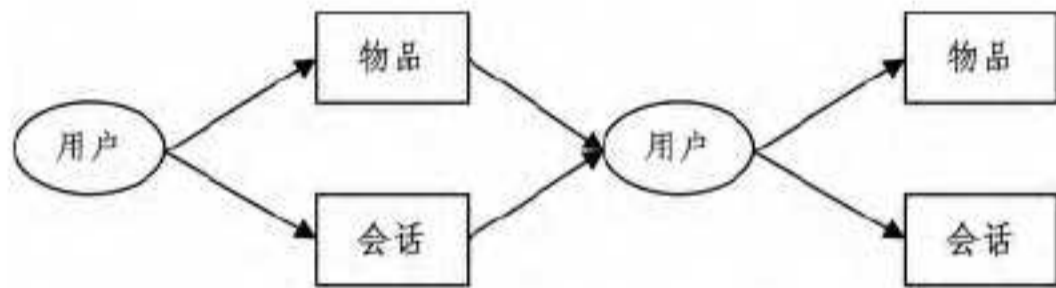


图2 最短路径图

4.2 复杂性分析

PFA 是基于图中的广度优先搜索, 并不涉及迭代。在 STG 子图中, 偏好只在与源节点距离小于或等于 3 的节点中传递。PFA 的主要计算复杂度来自于偏好传递的最后一步, 也就是从长度 2 的节点到长度 3 的节点。例如, 在无时间信息的 PFA 中, 长度 2 的节点是用户节点, 长度 3 的节点是物品节点。在最坏情况下, 长度 2 的节点包含所有用户节点, 然后偏好传递到所有物品节点, 在这里所有的用户节点都参与偏好传递, 所以复杂度是 $O(e \cdot |U|)$, e 是路径的总数, $|U|$ 是用户节点数目。对于有时间信息的 PFA, 由于引入了会话节点和相应的会话_物品路径, 假定它的路径数是无时间信息 PFA 的路径的数 k 倍, 因此它的复杂度是 $O(k \cdot e \cdot |U|)$ 。

从前面的复杂度的分析, 比较 PFA 和 UserKNN(基于用户的协同过滤算法)/ItemKNN(基于物品的协同过滤算法)的复杂度时, 我们发现无时间信息 PFA 的复杂度等于 UserKNN 的复杂度。由于路径增加, PFA 相比于无时间信息 PFA 较慢(k 是常数, 一般取值为 4)。

5 实验结果

5.1 数据集及评价指标

本文的目标是基于网络中丰富的隐性数据做出精确的 Top-N 推荐, 所以采用了两个历史数据集 CiteULike 和 Delicious。数据集 CiteULike 包含 52689 个用户, 1793954 个物品和 2119200 用户_物品对。CiteULike 数据集是非常稀疏的, 很多网页只有一个用户关注。首先除去少于 5 个关联用户的物品, 然后除去关联物品少于 5 个的用户后得到一个稠密数据集, 它包含 4607 个用户、16054 个物品和 109346 个用户_物品对, 数据集稀疏性为 99.85%。

Delicious 基于 Web 服务社交书签, 用于存储、分享和发现其它 Web 书签。Delicious 数据集也是非常稀疏的。将物

品被用户标记少于 10 个和用户打过标签物品少于 4 个的用户除去。由此产生一个稠密数据集, 包含 8861 个用户、3257 个物品和 59694 个用户_物品对。这个数据集的稀疏性为 99.79%。

在数据集中, 和时间相关的还有用户活跃度和物品生命周期。用户活跃度是用户活跃在这个系统的天数, 然而物品的生命周期是物品被一些用户查看或标记的天数。在 CiteULike 数据集中, 用户平均活跃度是 313 天, 物品的平均生命周期是 701 天。在 Delicious 数据集中, 用户平均活跃度是 138 天, 物品的平均生命周期是 309 天。

本文采用准确率评测推荐结果。将数据集分割成训练集和测试集。当生成推荐结果时, 本文为每个用户在时刻 t 生成一个 $N(N=10)$ 的物品列表, 并命名为 $R(u, t)$ 。如果测试物品出现在推荐列表中, 那么推荐成功。准确率是通过如下公式计算的:

$$Precision@ = \frac{\sum |R(u, t) \cap T(u)|}{\sum |R(u, t)|} \quad (7)$$

式中, $R(u, t)$ 是用户 u 在时刻 t 的 Top-N 物品推荐集, $T(u)$ 是用户 u 在时刻 t 访问的物品。

本文实验将对两个经典算法: UserKNN 和 ItemKNN。

5.2 基于无时态数据的参数 ρ

首先将实验数据呈现在用户_物品矩阵中(没有时间信息)。原因有二: 1) 系统参数 ρ (式(4)) 控制传递过程中节点出度的影响, 然后在接下来的试验中确定它; 2) 实验结果可以作为基线来判断时间数据是否能提高推荐系统的精度。

对于没有时间信息的用户_物品数据, 传递因子 ψ 如式(5)定义:

$$\psi(v, v') = \frac{1}{|out(v)|^\rho} \quad (8)$$

在 CiteULike 和 Delicious 数据集上, 比较无时间信息 PFA 和算法 ItemKNN 和 UserKNN 的准确率, 结果如图 3 和图 4 所示。

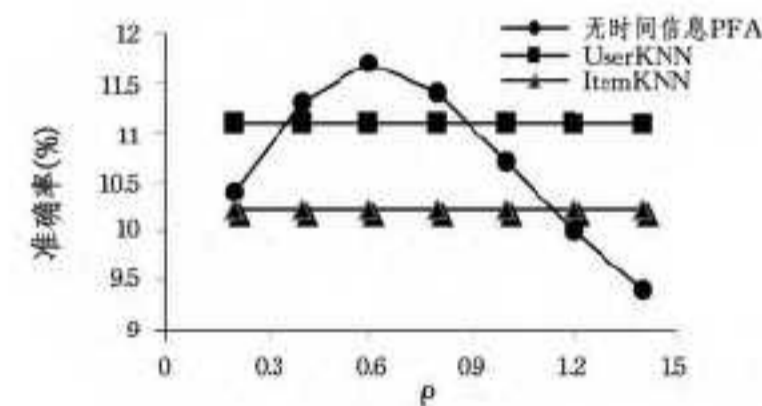


图3 在 CiteULike 数据集中, 参数 ρ 对无时间信息 PFA 的影响

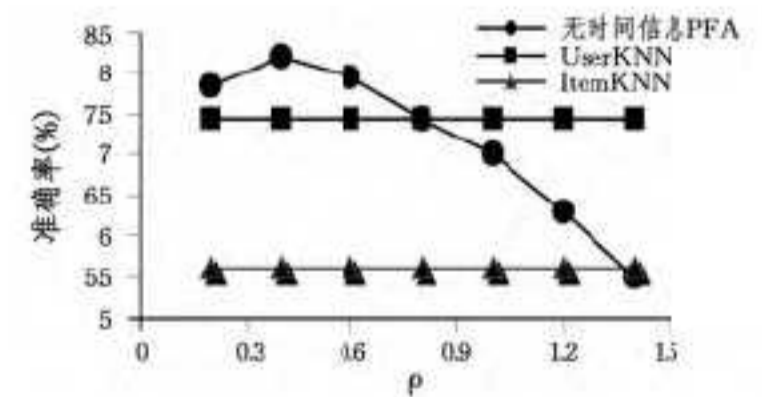


图4 在 Delicious 数据集中, 参数 ρ 对无时间信息 PFA 的影响

实验结果表明参数 ρ 对准确率有明显影响, 这表明偏好在传递过程中, 并不是正比于该节点的出度。在 CiteULike 数据集中, 当 ρ 为 0.6 时, 无时间信息 PFA 的准确率最高且大于其它两种算法准确率。而在 Delicious 中, 当 ρ 为 0.4 时, 无时间信息 PFA 的准确率最高且大于其它两种算法准确率。在两个数据集中 ρ 都小于 1, 这表明偏好传递到下个节点时

会被放大。

实验中,在 CiteULike 数据集中设置 ρ 为 0.6,在 Delicious 数据集中设置 ρ 为 0.4。

5.3 基于时态数据的参数 β 和 η 实验结果

在这部分中,数据具有时间信息。假定在 CiteULike 中最优时间单元是 7 天,而在 Delicious 中是 1 天,这是由于文件和网页之间的差异造成的。根据时间信息建立基于〈用户,物品〉和〈会话,物品〉数据的 STG。两数据集的 STG 的详细信息在表 2 中说明。

表 2 两数据集详细信息

	CiteULike	Delicious
用户	4607	8861
物品	16054	3257
会话	39939	45463
边	437384	238776

接下来研究参数 β 和 η 的影响,然后比较 3 个算法 (PFA、TItemKNN 和 TUserKNN) 的准确率。

5.3.1 参数 β 的影响

本小节分析参数 β (式 (3)) 在时效推荐中的影响。在 PFA 中, β 控制了物品节点相对于会话节点的输出偏好比例。 $\beta=0$ 表示推荐结果只受用户短期偏好影响; $\beta=1$ 表示推荐结果只受用户长期偏好影响。 β 平衡了长期和短期在两种算法中的影响, β 越大,长期偏好影响越大。

实验中将参数 η 假定为 1 即 $\eta_l = \eta_s$,即在测试物品间相似度上平衡了长期和短期偏好影响。PFA 算法的准确率随 β 变化情况如图 5 和图 6 所示。算法 TItemKNN 和 TUserKNN 没有参数 β ,所以它们的准确率被描绘成一条直线。结果表明在两个数据集中,当 $\beta \in [0.1, 0.9]$ 时 PFA 优于其它算法。结果还表明忽视长期偏好 ($\beta=0$) 和短期偏好 ($\beta=1$) 都不能得到最好的结果。最好的结果是将长期和短期因素结合在一起。在 PFA 算法中, β 相当稳定(任取 $\beta \in [0.1, 0.5]$ 所得结果准确率相近)。因此,在接下来实验中,取 $\beta=0.5$ 。

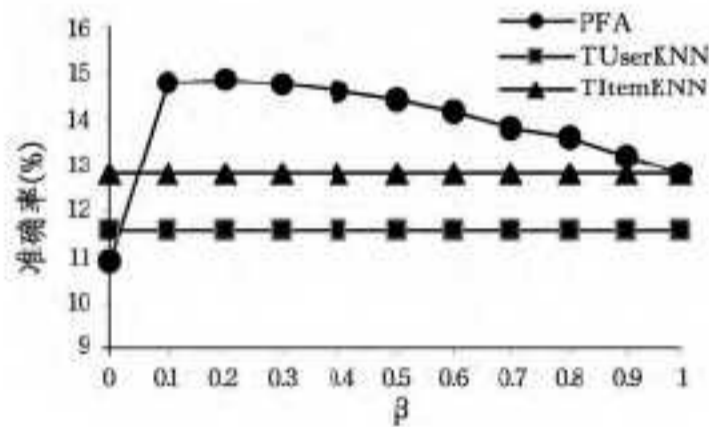


图 5 在 CiteULike 数据集中,参数 β 对 PFA 的影响

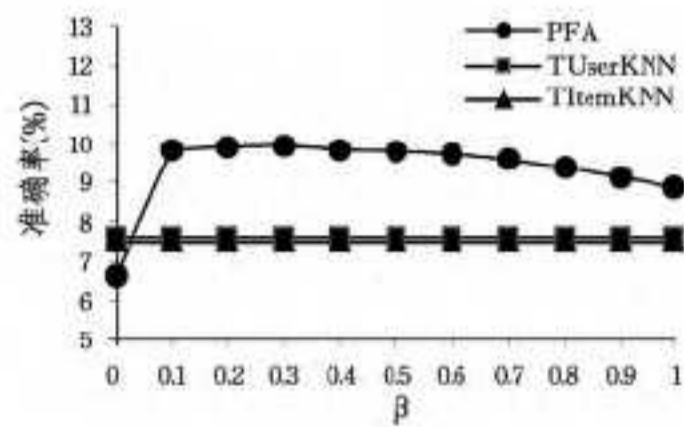


图 6 在 Delicious 数据集中,参数 β 对 PFA 的影响

5.3.2 参数 η 的影响

本小节分析在用户相似度计算方面参数 η (式 (5)) 对平衡长期和短期偏好的影响。 $\eta \rightarrow \infty$ 表示用户只和物品节点关联,用户_用户的相似度只决定于用户长期偏好; $\eta \rightarrow 0$ 表示用户只和会话节点关联,用户_用户的相似度只决定于用户短期偏好。图 7 和图 8 描绘算法准确率随 η 变化趋势。从实验结

果可以看出在两个数据集中, $\eta=1$ 时 PFA 算法的准确率最高且优于其它算法,这也表明用户长期和短期兴趣都对测量物品相似度有非常重要的作用。接下来实验中取 $\eta=1$ 。

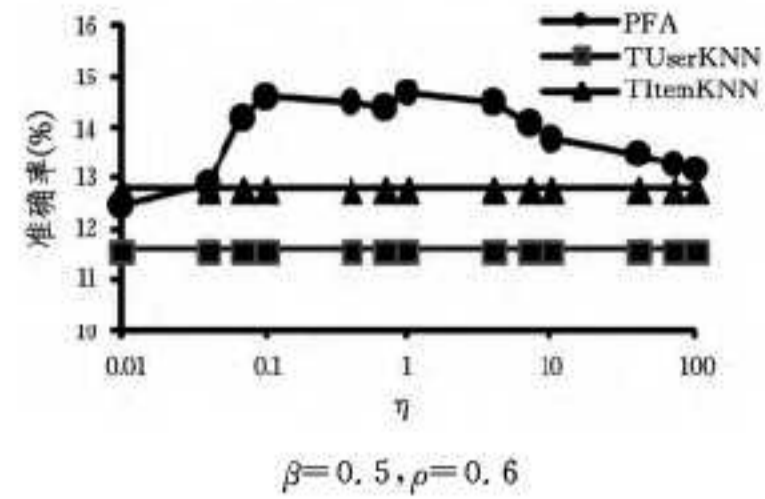


图 7 在 CiteULike 数据集中,参数 η 对 PFA 的影响

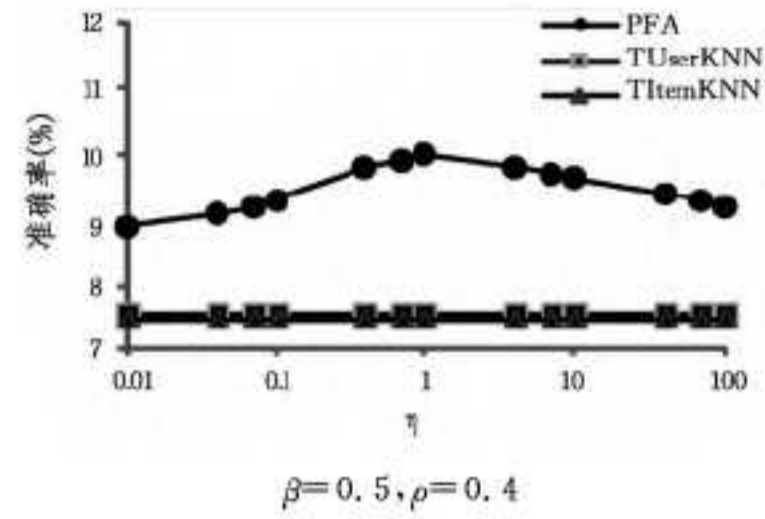


图 8 在 Delicious 数据集中,参数 η 对 PFA 的影响

5.3.3 时间段的大小

实验中把时间分割成段,然后将物品和与其相关的时间段结合成会话节点。选择时间段的大小时,需要考虑两个准则:一是推荐精度,这意味着需要选择一个合适的时间段大小去生成一个更准确的推荐;二是算法复杂性,当时间段比较小时,会话节点和边的数目就会非常庞大,这会增加计算复杂度。所以当在两个大小不同的时间窗口进行推荐时,得到相似的准确率,一般选择时间段大的那个。为了得到时间段大小对推荐精度的影响,实验中,在 CiteULike 数据集中设置 $\beta=0.5, \eta=1, \rho=0.6$,在 Delicious 数据集中设置 $\beta=0.5, \eta=1, \rho=0.4$,得到结果分别如图 9 和图 10 所示。

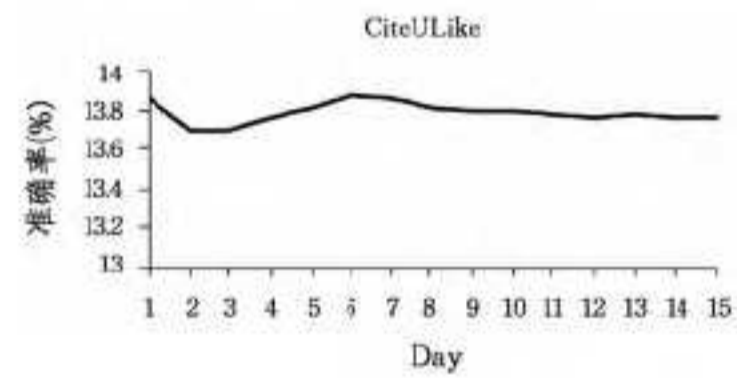


图 9



图 10

从实验结果可以看出,在 CiteULike 数据集中,时间窗口大小在半个月之内,算法都有相似的精度,最佳大小是 6 到 7 天(一个星期左右),这也就是本文在前面实验中设定的。对于 Delicious 数据集,时间窗口大小在 7 天之内,推荐都有相似的精度,最佳大小是 1 天。结果表明 PFA 的精度对时间窗口大小选择不敏感。只要算法将窗口大小设置在一定范围内,都会产生相近的结果。

5.4 实验结果

主要比较上述算法在时效 Top-N 推荐的总体精度。

表 3 展示在最优参数情况下,上述算法在 CiteULike 数据集上的最好精度。结果显示 TUserKNN 精度最低,而 PFA 精度最高。以 TItemKNN 为基准,PFA 比 TItemKNN 提高 15.63%。表 4 展示在最优参数情况下,上述算法在 Delicious 数据上的最好精度。在这个数据集上,TItemKNN 和 TUserKNN 的结果接近,相比于 TItemKNN,PFA 提高了 35.07%。可以看出 PFA 优于所有其它时效算法,这证明 PFA 在准确推荐中平衡了用户长期和短期偏好影响。

表 3 最优参数情况下所有算法在 CiteULike 数据上的最好精度

算法	准确度	提高
TItemKNN	12.86%	—
TUserKNN	11.62%	-9.64%
PFA	14.87%	15.63%
无时态 PFA	11.72%	

表 4 最优参数情况下所有算法在 Delicious 数据上的最好精度

算法	准确度	提高
TItemKNN	7.47%	—
TUserKNN	7.58%	1.47%
PFA	10.09%	35.07%
无时态 PFA	8.02%	

结束语 用户偏好一般受长期和短期因素影响。捕获和利用这些因素能够有效提高时效推荐的性能并具有很大的挑战性。本文基于三分图的模型,捕获随时间推移用户的长期和短期偏好因素,在基于会话的时态图(STG)的基础上,设计了路径融合算法 PFA(Path Fusion Algorithm)并将其应用于时效推荐。在真实历史数据集上进行的实验表明本文提出方

法的有效性较传统方法有明显提高。进一步研究的工作包括在 STG 的基础上进行用户和物品的聚类,以及用更灵活的方式去确定时间窗口的大小和会话节点选择的策略,以使该方法应用到更多的领域并得到良好的推荐效果。

参考文献

(上接第 376 页)

- [5] Nguyen P, Shparlinski I. On the insecurity of the elliptic curve digital signature algorithm with partially known nonces[J]. Designs, Codes and Cryptography, 2003, 30(2): 201-217
- [6] Liardet P Y, Smart N P. Preventing SPA/DPA in ECC systems using the Jacobi form[C] // CHES2001. LNCS 2162, 2001: 391-401
- [7] Oswald E, Aigner M. Randomized addition-subtraction chains as a countermeasure against power attacks[C] // Proc. CHES2001. LNCS 2162, 2001: 39-50
- [8] Zhang N, Chen Z X, Xiao G Z. Efficient elliptic curve scalar multiplication algorithms resistant to power analysis[J]. Information sciences, 2007, 177: 2119-2129
- [9] Okeya K, Takagi T. The width-wNAF method provides small memory and fast Elliptic scalar multiplications secure against side channel attacks [C] // Topics in Cryptology (CT-RSA 2003). LNCS 2612, 2003: 328-343
- [10] Joye M, Quisquater J J. Protections against differential analysis for elliptic curve cryptography[C] // Proc. CHES 2001. LNCS 2162, 2001: 3402-410
- [11] Smart N P. The Hessian form of an elliptic curves[C] // Proc. CHES2001. LNCS 2162, 2001: 3118-125
- [12] Billet O, Joye M. The Jacobi model of an elliptic curve and side-channel analysis[C] // Applied Algebra, Algebraic Algorithms and Error-Correcting Codes(AAECC 2003). LNCS 2643, 2003: 34-42
- [13] Chevallier-Mames B, Ciet M, Joye M. Low-cost solutions for preventing simple side-channel analysis; side-channel atomicity [J]. IEEE Transaction on Computers, 2004, 53(6): 760-768
- [14] Al-Somani T F, Amin A. An Efficient High Performance Scalar Multiplication Method with Resistance against Timing Attacks [C] // IEEE/ACS Int. Conf. on Computer Systems and Applications(AICCSA 2008). Doha, 2008: 860-865
- [15] 王敏, 吴震. 抗 SPA 攻击的椭圆曲线 NAF 标量乘实现算法[J]. 通信学报, 2012 33(Z1): 228-232
- [16] Hankerson D, Menezes A, S. Vanstone. Guide to elliptic curve cryptography[M] // Professional Computing Series. Springer-Verlag, 2004
- [17] Okeya K, Schmidt-Samoa K, Spahn C, et al. Signed binary representations revisited[C] // Advances in Cryptology (CRYPTO'04). LNCS 3152, 2004: 123-139
- [18] 邬可可, 李慧云. 一种高效的可防御侧信道攻击的椭圆曲线标量乘法方法[J]. 先进技术研究通报, 2010, 4(5): 52-58
- [19] Jebril I. H, Salleh R, Al-Shawabkeh M. Efficient Algorithm in Projective Coordinates for EEC Over $GF(2^n)$ [J]. International Journal of The Computer, the Internet and Management, 2007, 15: 43-50