基于用户访问序列的实时网页推荐研究

Real – Time Web Page Recommendation Research Based on User Access Sequence

杨正余　王卫平　（中国科学技术大学信息管理与决策科学系 安徽合肥 230026）

摘　要： 应用序列模式挖掘的网页推荐系统具有较高的准确率。但是，目前广泛应用的基于树型结构的序列模式挖掘在页面推荐前需花费大量时间来统计历史访问页面的访问次数，降低了推荐效率。本文介绍了一种智能化网页推荐系统模型，该方法无需统计每个页面的访问次数，避免了重复访问数据库，且利用用户即时访问的滑动窗口，直接在模式树中搜索相匹配的访问规则，加快了推荐速度，较好地满足页面推荐实时快速的要求，最后试验表明其具有较好的推荐效果。

关键字： web 访问序列　推荐系统　web 个性化　网页推荐　序列模式挖掘

1　引言

随着因特网的快速发展，存在于互联网上的数据呈爆炸式地增长，而这些数据背后隐藏着许多重要的有价值的信息，如何找到这些有价值的信息，为网络用户提供个性化的服务，成了当前 web 个性化领域一个重要研究课题。目前，基于 web 使用挖掘的 web 推荐技术发展迅速，通过用户的访问日志可以获得页面的点击次数、页面停留时间、页面访问顺序等信息，分析之后可以理解用户的行为和需要，得到用户群体的访问行为模式，从而为用户定制推荐页面。在基于 Web 使用挖掘的推荐系统中，较为常用的方法有关联规则、分类聚类和序列模式挖掘等。但各种方法都有着不同的特点：

(1) 关联规则挖掘技术预测用户的浏览模式在网页个性化研究领域引起了多方面的关注。但是，其预测的规则和用户的浏览行为之间的匹配率较低，其结果并不令人满意。

(2) 分类聚类的 web 个性化技术在预测用户浏览路径时虽然覆盖率(Coverage)较高，但其要花费大量时间来计算访问路径之间相似度，延缓了推荐速度，且对网站服务器的性能要求也较高。

(3) 由于考虑了页面的访问顺序，从考察的信息量上来说，序列模式挖掘较其它两种方法要多，所以在推荐的质量上，序列模式能够达到较高的准确性，而聚类模式较低，关联规则介于两者之间[1]。

由于序列模式挖掘在网页推荐中具有较为明显的不同与优点，其也受到很多学者的关注，但是，目前广泛应用的基于树型结构的序列模式挖掘在页面推荐前需花费大量时间来重复访问数据库，统计历史访问页面的访问次数，从而大大降低了页面推荐速率[2,3]。基于序列模式挖掘在网页推荐上的优点，本文利用序列模式的 web 挖掘技术，综合利用人工智能，设计了一个推荐系统模型，并在此系统中应用了一种新的个性化页面推荐方法，其无需统计页面访问次数，故仅需一次扫描数据库，效率明显得到提高，另外算法利用用户即时访问的滑动窗口，直接在模式树中搜索相匹配的访问规则，加快了推荐速度，所以能够实时快速地向用户推荐访问页面。

2　页面推荐系统模型

自适应网站是未来网站的发展趋势，其核心是智能化的推荐系统，为了满足其功能需要，本文给出了如图 1 所示的架构，并对推荐系统的各个部件进行优化，其主要由两个子系统组成，分别为在线推荐子系统和离线挖掘子系统，其中数据预处理模块和序列模式挖掘模块属于离线挖掘子系统，而推荐规则产生模块属于在线推荐子系统，这种设计方法可以最大限度地减少服务器的负载，并提高系统的执行效率。具体推荐

过程可描述为当一个用户访问网站时，

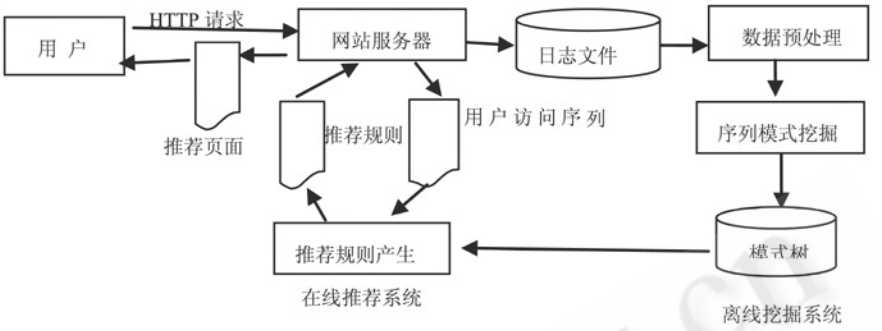

图1　页面推荐系统模型

网站服务器跟踪用户会话，记录下用户访问路径，然后存储在访问日志中。日志中的数据是按照所有用户的访问时间顺序存放的，而且由于多种原因，原始数据并不适合直接挖掘，必须将日志数据经过预处理后才能使用。数据预处理模块定期从访问日志中抽取数据，然后把预处理过的数据提交给序列模式挖掘模块，序列模式挖掘模块根据最新的访问数据定期增量更新访问模式树，并将结果保存下来。在线推荐子系统根据当前用户的访问序列，并结合推荐算法，判断该用户此时极可能访问的页面集合，并反馈给网站服务器。然后，网站服务器再根据用户实际点击的链接，在目标网页上增加可能即将访问的页面集再反馈给用户。

3　基于用户访问序列的页面推荐

3.1　基本概念

假设 P 是经过预处理后的访问页面集合。一条 Web 日志文件中的页面访问序列是按照访问时间先后顺序而排列的，可以表示为 $S = P_1, p_2, p_3, \cdots, p_n$（$p_i \in P, 1 \leqslant i \leqslant n$，n 称为访问序列的长度）。长度为 n 的序列也叫做 n 访问序列，在一个访问序列 S 中，当 $i \neq j$ 时，可以有 $p_i = p_j$。当且仅当存在 $1 \leqslant i_1 < i_2 < \cdots < i_n \leqslant n, p_i' = p_i, 1 \leqslant j \leqslant m$ 时，访问序列 $S' = P'_1, p'_2, p'_3, \cdots, p'_n$ 是 S 的子序列，s 称为 S'的超序列。当且仅当 S'是 s 的子序列且 $S' \neq S$ 时，S'才是 S 的真子集。

Web 访问序列的集合 $WASD = \{S_1, S_2, \cdots, S_m\}$ 称为 web 访问序列数据库，S_i 是访问序列（$1 \leqslant i \leqslant m$）。某一访问子序列 S'在 WASD 上的支持度 sup（S'）就是 WASD 中包含 S'的访问序列数占所有访问序列数的比例，且有：

$$sup(S') = \frac{|\{S' | S' \subseteq S, S \in WASD\}|}{m}$$

假定一个支持度阈值 δ，如果 sup（S'）$\geqslant \delta$，则序列 S'叫做 δ 模式或者 WASD 的频繁访问模式。Web 访问模式推荐就是给出一个 web 访问序列数据库 WASD 和支持度阈值 δ，根据用户的即时访问会话子序列，向用户推荐满足支持度阈值的页面。

3.2　页面推荐算法

整个算法分两个过程，第一步根据用户的访问序列数据库，构建用户访问模式树（User Access Pattern – tree，UAP – tree），第二步根据用户访问模式树实时地向用户推荐访问页面。

3.2.1　用户访问模式树的构建

在这里 UAP – tree 由两部分组成，分别是头表和访问模式树。其定义如下：

（1）由一个根节点 root，根结点的访问前缀子树（prefix subtree）和一个项目头表（header link list）组成。

（2）前缀子树的每个节点由 itemname，count，nodelink 和 childlink 四个域组成。其中 itemname 表示结点（每一访问页面代表一访问结点）名，count 表示通过该结点的路径个数，nodelink 指向下一个具有相同 itemname 的结点，若其后无同名结点，则该域为 null。Childlink 指向该结点的孩子结点，若该结点为叶子结点时，该项为 null。

（3）头表由三个域组成，分别为 itemname，headlink，taillink，其中 itemname 表示访问结点名，headlink

表示指向用户访问模式树中与 itemname 相同名的第一个结点，taillink 表示指向用户访问模式树中与 itemname 相同名的最后一个结点。

具体构造过程：

UAP – tree 构建算法：

输入：访问序列数据库 WASD

输出：访问模式树 T 和项目头表 H

1) create T，并且令 T. nodelink = null；

2) create header link list H，包含 itemname，headlink 和 taillink 域；

3) create a currentnode newT；

4) For 对于每一个访问序列 $S_i = = P_1, _{p2}, p_3, \cdots, p_n \subseteq$ (WASD do

{令 newT = T；//当前结点指向根结点

for each $P_i \in S_i$ do {

if 当前结点的孩子结点 child. itemname = p_i then

set child. count = child. count + 1；newT = child；

continue ；

//说明头表中有该结点，无需再建立头表

else 创建一个新结点 newnode；newnode. itemname = p_i；

newnode. count = 1；

令 newT = newnode；

AddToH(H，newnode) {for 每一个头表项 HItem do

{if HItem. itemname = newnode p_i then

p_i. taillink. nodelink = newnode；p_i. taillink = newnode；

continue；

else 创建一个新头表项 newHItem

令 newHItem. itemname = p_i；

newHItem. taillink = newnode；

newHItem. headlink = newnode；}}}

endfor

newT = T；}

endfor

5) return H，T；

3.2.2 基于用户访问模式树的推荐算法：

对于 l 长度序列的在线访问模式，考虑到 web 页面被访问是有先后顺序的，而不同的用户的访问浏览路径不尽相同，所以这里 l 的确定我们采用“滑动窗口”的概念[5]。滑动窗口是用户最近访问路径的一个子集，如果事先规定滑动窗口的大小为固定值，则滑动窗口内的页面会随用户的浏览访问而不断改变。例如，如果规定滑动窗口的大小为 3，对于访问序列 < A，B，C，D > 的滑动窗口页面序列为 < B，C，D >，如果用户紧接着又访问了 E，窗口序列就变为 < C，D，E >。具体的推荐算法实现过程：

输入：用户访问模式树的头表，H；滑动窗口 W；最小支持度阈值 δ；

输出：W 的推荐页面 recommendpages；

1) 令 p = W[1]；NodeSet = ∅；recommendpages = ∅；

// W[1] 为 W 中的第一个页面；

2) for each Hitem 头表 H do

if Hitem. itemname = p then

FirstNodeSet(p. headlink，p. taillink)

//搜索树，NodeSet 为有相同父结点的结点集

{for each p. childnode do

NodeSet = NodeSet ∪ p. childnode；

If p. childlink = p. taillink；

return NodeSet；

else continue；}

3) for(l = 2；l < = n；l + +) //n 为 W 的窗口大小

NextNodeSet(NodeSet，W，l)

//寻找与 W[l] 相匹配的结点

{for i = 1 to m do

//m 为 NodeSet 的长度

if tempNode[i] = = W[l]

for each NodeSet[i]. childlink do

NodeSet = NodeSet ∪ childnode；

return NodeSet；}

4) 令 tempNodeSet = NodeSet；

for i = 1 to m

令 count = 0；

for(j = m；j > 0；j - -)

{if NodeSet[i] = = tempNodeSet[j] then

count = count + tempNodeSet[j]. count；

delete tempNodeSet[j] from temNodeSet；}

if count≥δ|WASD| then
recommendpages = recommendpages ∪ NodeSet[i]；
// recommendpages 即为最终的推荐集
5) return fail；//无推荐页面

4 试验评价

本文采用 Mobasher 提出的评价测度来分析推荐质量，包括准确率(Precision)，覆盖率(Coverage)和 F 测度(F－measure)[5]。准确率指在推荐的内容中用户喜欢的项占所有推荐项的比例，而覆盖率指在推荐的内容中用户喜欢的项占所有用户喜欢的项的比例，F 测度是将两者综合起来从整体上考虑系统的推荐质量的度量值。若用户喜欢的页面集合表示为 LS，系统推荐给用户的页面集合为 RS，推荐的页面集合中用户喜欢的页面集合为 R_LS，根据定义，三个测度的计算公式为：

(1) Coverage = |R_LS|/|LS|

(2) Precision = |R_LS|/|RS|

(3) F = 2 * Coverage * Precision/(Coverage + Precision)

试验采用的访问日志数据来自网站[7]。经过数据清洗、用户识别、事务识别和路径完善等数据预处理步骤后，最后得到 13745 个用户访问会话和 683 个 URL 访问记录，抽取大约整个数据集的 2/3 作为训练数据集，用以建立用户访问模式和推荐结果分析集，另外的大约 1/3 数据作为测试数据。为了得到实验结果，将访问支持度阈值 δ 分别设置为 0.1、0.2 和 0.3，而滑动窗口 n 的大小设置为 2、3 和 4，分别测试覆盖率、准确率和计算 F 的大小。试验结果如表所示：

表 1　测试结果：|w| =2

δ	Coverage	Precision	F
0.1	0.79	0.51	0.61
0.2	0.71	0.55	0.62
0.3	0.66	0.59	0.62

表 2　测试结果：|w| =3

δ	Coverage	Precision	F
0.1	0.69	0.53	0.60
0.2	0.63	0.58	0.60
0.3	0.57	0.66	0.61

表 3　测试结果：|w| =4

δ	Coverage	Precision	F
0.1	0.58	0.63	0.61
0.2	0.56	0.65	0.60
0.3	0.51	0.71	0.59

可以从结果看出，当滑动窗口大小一定时，准确率随着阈值的增大而增大，覆盖率随着支持度阈值的增大而减小，这是因为随着支持度阈值增大，包含在推荐页面集中用户喜欢的页面就越多，从而被用户点击的可能性也越大，准确性也就越高。支持度阈值增大时，虽然准确率提升，但满足阈值的页面数将变少，能匹配的规则也相应减少，故覆盖率下降。相类似的原因，可以看出，当支持度阈值一定时，随着窗口数的增大，用户访问的准确率提升，而覆盖率却下降，但在几种支持度阈值情况下，总体性能变化不大。

5 结束语

为用户推荐感兴趣页面，满足用户需求，对提高各种商务网站的点击率有着重要意义。论文首先设计给出了一个经过优化的推荐系统模型，其功能结构能够很好满足自适应网站的建设。在此系统中，应用了一种新的为在线访问用户实时推荐 web 网页的方法。其只需一次扫描访问序列数据库就可将用户的历史会话序列压缩存储在 UAP－tree 上，由于头表中省去了页面访问次数域，所以避免重复访问数据库。接着文章给出了页面推荐算法，该算法根据用户在线访问序列的滑动窗口，根据用户最近的页面序列，向用户推荐符合规则的页面集，所以能够很好地满足在线实时页面推荐的要求。

参考文献

1 Mobasher B, Dai H, Luo T, Nakagawa M. Using Sequential and Non－Sequential Patterns in Predictive Web Usage Tasks. Proceeding of ICDM2002, 2002.

2 X Q Tan, M Yao, J K Zhang. Mining Maximal Frequent Access Sequences Based on Improved WAP－tree. Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA' 06), 2006.

(下转第 13 页)

(上接第 53 页)

3 M. El – Sayed, C. Ruiz, E. A. Rundensteiner. FS – miner: efficient and incremental mining of frequent sequence patterns in web logs. ACM, 2004.

4 Mobasher B, Dai H, Luo T. Discovery of aggregate usage profiles for web personalization[C]. Proceedings of the ACM SIGKDD, 2000: 142 – 151.

5 Mobasher B, Dai H, Luo T, et. al. Effective Personalization Based on Association Rule Discovery from Web Usage Data[C]. Proceedings of ACM Workshop on Web Information and Data Management (WIDM), 2001: 103 – 112.

6 HAN JW, KAMBER M. Data Mining: Concepts and Techniques [M]. San Mateo, CA: Morgan Kaufmann, 2000.

7 Depaul CT1 web usage mining data[OL]. http://maya.cs.depau1.edu/~classes/etc584/resource.html