文章编号: 1001-0920(2007)03-0284-05

一种新的 Web 用户群体和 URL 聚类算法的研究

宋江春, 沈钧毅
(西安交通大学 电子与信息工程学院, 西安 710049)

摘 要: 提出一个基于 Web 日志的 Web 用户群体和站点 URL 聚类算法. 使用用户浏览行为描述和用户浏览时间离散化方法建立了 Web 站点的用户事务矩阵, 并在此基础上对 Web 用户群体和站点 URL 进行聚类. 由于在聚类过程中同时考虑了用户对 URL 的浏览时间和访问次数, 使算法的精度和效率都大大提高. 同时, 该算法能较好地处理类间重叠问题, 使算法具有较好的实用性. 最后对算法的有效性和可伸缩性进行了研究.

关键词: Web 使用挖掘; 用户浏览模式; 用户访问矩阵; 用户事务聚类; 站点 URL 聚类
中图分类号: TP391.11 **文献标识码**: A

Research on a new clustering algorithm of Web user communities and Web site's URLs

SONG Jiang-chun, SHEN Jun-yi
(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. Correspondent: SONG Jiang-chun, E-mail: sjchun@163.net)

Abstract: By using new methods which are based on Web user's browsing behavior characterization and user's viewing time discretization, a new clustering algorithm for Web user communities and Web site's URLs is proposed. Web user access matrixes are set up on the preparation of Web logs. By considering user's viewing time and number of hits to Web site's URLs simultaneously, the accuracy and efficiency of the clustering algorithm are increased. The improved algorithm could solve the problem of the partial overlap bewteen clusters, which makes the algorithm more practical. The effectiveness and the scalability of the algorithm are studied through the experiments.

Key words: Web usage mining; User browsing pattern; User access matrix; User session clustering; Web site URL clustering

1 引 言

Web 使用挖掘是研究用户 Web 浏览行为的技术和工具, 理解访问者的浏览兴趣是提高 Web 服务质量和改善站点结构设计的重要环节. 通过分析和探究用户访问情况中的规律, 可以识别电子商务的潜在客户, 增强对最终用户的因特网信息服务的质量和交付, 并改进 Web 服务器系统的结构和性能[1,2]. Web 使用挖掘技术的一个重要的研究方向是 Web 用户事务聚类和站点 URL 聚类, 即通过用户对网站的使用信息 ——Web 日志的处理和研究, 得到具有相似访问兴趣的用户群体和用户共同感兴趣的站点 URL, 据此可以判别和调整站点的结构并进行个性化服务[3,4].

对于用户事务聚类, Fu 等[5] 采用分层聚类算法 (BIRCH), 利用一种概化的方法来聚类 Web 使用事务. Shahabi 等[6] 采用 K-means 算法来聚类用户事务, 它引入路径特征空间和路径角度的概念, 访问向量的元素是用户事务中页面出现的次数, 相似度的度量采取两个访问向量角度的 cos 值.

对于站点 URL 聚类, 之前主要采用基于内容的方法, 目前研究趋势是采用基于用户使用的方法, 认为经常被用户同时访问的 URL 是紧密相关的. 苏中等[7] 采用了基于密度的递归聚类方法, 在聚类过程中动态地改变聚类参数以改善 URL 聚类结果. Perkowitz 等[8] 用 PageGather 聚类算法在 Web 站点上寻找相关页面 URL 的集合, 首先创立相似

收稿日期: 2005-11-28; **修回日期**: 2006-02-03.
基金项目: 国家自然科学基金项目(60173058).
作者简介: 宋江春(1962—), 男, 成都人, 副教授, 博士生, 从事模式识别、智能信息控制等研究; 沈钧毅(1939—), 男, 江苏扬州人, 教授, 博士生导师, 从事模式识别、智能信息控制等研究.

矩阵,矩阵的元素是从 Web 日志得到的页面间共同被访问的频度,然后在该矩阵的基础上进行 URL 聚类.

以上研究均存在一些不足,首先在聚类的相似性度量方面,单纯地以浏览时间或访问次数来度量,对于 Web 站点这种复杂的情况而言,该聚类是不够准确的. 另外,他们均采用传统的聚类技术,即把每个待辨识对象严格地划分到某个类中,不能处理类间重叠问题.

本文充分考虑了上述问题,在对 Web 日志进行预处理后,建立以用户浏览离散化时间为度量的 Web 站点用户浏览矩阵,并在此基础上,利用用户浏览矩阵的行列向量的距离进行初始聚类;然后利用用户浏览次数定义的连接强度进行求精,从而得到最终聚类. 在聚类过程中同时考虑用户对 URL 的浏览时间和访问次数,使算法的精度和效率都大大提高. 最后通过实验对算法的性能进行了验证.

2　Web 站点用户访问矩阵

Web 服务器日志包括访问日志、引用日志和代理日志. 对这些日志进行合并和剔除后,以 $L = \langle$ip, uid, url, time$\rangle$形式表示 Web 服务器日志. 其中 uid, ip, url, time 分别为用户 ID、用户 IP 地址、用户请求的 URL 和相应的浏览时间. 然后对它进一步处理,以反映用户在某一段时间内的浏览行为.

2.1　用户浏览行为描述

通过对日志的处理,可以得到一个用户在一段连续时间范围内的访问 URL 序列,称为用户事务. 具体的方法有最大向前路径法、最大时间长度法和最大时间间隔法等[9,10]. 本文利用最大时间间隔法来得到用户事务集合,具体定义如下:

定义 1(用户事务)　设 UT 为用户事务集合,对于 $\forall ut_i \in UT$,

$$ut_i = \langle IP_i, UID_i, \{(l_i.URL, l_i.time, l_i.hits)\}^m\rangle. \quad (1)$$

其中: $l_i \in L$, m 表示 Web 站点的 URL 数, $l_i.time$ 表示到目前为止用户 UID_i 在页面 $l_i.URL$ 上的浏览时间, $li.hits$ 表示浏览次数.

聚类时,首先考虑用户事务在站点 URL 上的浏览时间,将离散化技术应用到用户浏览时间的表示上,将时间属性划分为区间,用区间标号来代替实际时间值.

按照用户在 URL 上的浏览时间将 Web 访问进行离散化. 离散值和浏览时间的关系如表 1 所示.

通过这种浏览时间离散化表示方法,认为如果用户在 URL 上的停留时间小于 5 s,则该 URL 是导航页而不是内容页,应该删除,用户在 URL 上的浏览时间即使再长,也只有离散化时间 3. 这种方法可有效避免在进行用户事务相似性度量时,采用连续时间造成的由于用户浏览时间过小或过大而导致的聚类结果畸变问题.

表 1　访问情况和离散值对照表

离散值	访问情况 / s
0	Time ≤5
1	5 < Time ≤60
2	60 < Time ≤180
3	180 < Time

注: Time 表示浏览时间.

2.2　Web 站点用户访问矩阵

定义 2(Web 站点)　一个 Web 站点就是一幅具有如下形式的有向图: $G = (N, N_p, E, E_p)$. 其中: N 为结点集; $N_p = \{\{(N_i.UID, N_i.time, N_i.hits)\}^n, N_i \in N\}$, $n \geq 1$ 为访问站点 N_i 的用户数, $i = 1, 2, \cdots, m$ 为 Web 站点的 URL 数,记录用户 UID 及其访问结点 N_i 的停留时间及访问次数,为结点属性集; E 为有向边集; $E_p = \{\{$Numedge of$\langle N_i, N_j\rangle\}, \langle N_i, N_j\rangle \in E\}$,记录有向边所在路径的编号,为有向边属性集. 如图 1 所示.

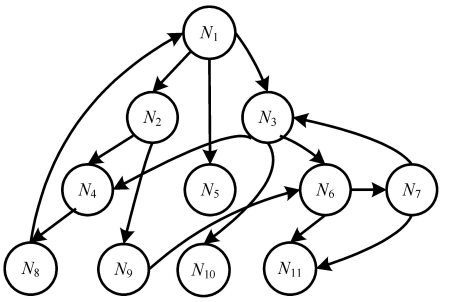

图 1　Web 站点的有向图表示

从有向图 G 的结点集 N 中可以得到该站点的所有 URL,从相应的结点属性集 N_p 中可获得访问每一个结点的 UID 及相应的浏览时间,据此建立如下所示的 URL-Session 关联矩阵.

定义 3(用户浏览矩阵)　根据定义 1,定义 2 和表 1 建立 Web 站点的用户浏览矩阵,即

$$T_{m\times n} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1j} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2j} & \cdots & t_{2n} \\ \cdots & \cdots & \ddots & \cdots & \ddots & \cdots \\ t_{i1} & \cdots & t_{i2} & t_{ij} & \cdots & t_{in} \\ \cdots & \cdots & \ddots & \cdots & \ddots & \cdots \\ t_{m1} & t_{m2} & \cdots & t_{mj} & \cdots & t_{mn} \end{bmatrix}. \quad (2)$$

其中: m 为站点 URL 的个数, n 为用户事务数, t_{ij} 为第 j 个用户事务对第 i 个 URL 的总的离散化浏览时间.

定义 4(用户点击矩阵) 根据定义 1 和定义 2 建立 Web 站点的用户点击矩阵,即

$$H_{m\times n}=\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1j} & \cdots & h_{1n}\\ h_{21} & h_{22} & \cdots & \cdots & h_{2j} & h_{2n}\\ \cdots & \cdots & \ddots & \cdots & \ddots & \cdots\\ h_{i1} & h_{i2} & \cdots & h_{ij} & \cdots & h_{in}\\ h_{m1} & h_{m2} & \cdots & h_{mj} & \cdots & h_{mn} \end{bmatrix}. \quad (3)$$

其中:m 为站点 URL 的个数;n 为用户事务数;h_{ij} 为第 j 个用户事务在一段时间内访问第 i 个 URL 的次数.

将用户浏览矩阵和用户点击矩阵统称为 Web 站点用户访问矩阵.

3 聚类算法

从定义 4 中看出,每一列向量 $T[\cdot,j]$ 表示所有用户对 URL"j"的访问情况,每一行向量 $T[i,\cdot]$ 表示用户"i"对该站点中所有 URL 的访问情况. 因此可以认为,列向量既代表了站点的结构,又蕴涵了用户共同的访问模式;行向量既反应用户类型,又刻画出用户的个性化访问子图. 于是,分别度量列向量和行向量的相似性,就能得到相似用户群体和相关 URL.

3.1 用户群体聚类算法

在进行相似用户事务聚类时,相似性度量是根据列向量之间的 Manhattan 距离来进行的,距离越小,相似程度越高. 本节首先根据用户浏览矩阵列向量之间的 Manhattan 距离进行聚类;然后根据用户点击矩阵定义的连接强度对类进行确认,得到最终聚类. 首先给出相关定义.

定义 5(Manhattan 距离) 向量 $T[\cdot,k]$ 和向量 $T[\cdot,p]$ 的 Manhattan 距离 M_d 定义为

$$M_d=(T[\cdot,k],T[\cdot,p])=\sum_{i=1}^{m}|t_{ki}-t_{pi}|. \quad (4)$$

聚类时,根据定义 5 计算列向量之间的 Manhattan 距离,建立列向量间的距离矩阵 $D_{n\times n}^{M_d}$,矩阵元素 d_{ij} $(1\le i\le n,\ 1\le j\le n)$ 表示第 i 个用户事务和第 j 个用户事务之间的 Manhattan 距离,D 是对称矩阵,对角线上的元素为 0.

对指定阈值 σ,若 $d_{ij}\in D_{n\times n}^{M_d}(1\le i\le n, i<j\le n)$,且 $d_{ij}<\sigma$,那么就将第 i 个用户和所有满足这个条件的第 j 个用户划分为一个类.

上述聚类过程仅考虑用户在某一 URL 的停留时间,而没有考虑用户对该 URL 的访问频率. 利用用户点击矩阵(3)来对聚类的结果进行确认. 首先给出如下的定义.

定义 6(连接强度) 设 T 是用户事务的集合,任意 $T[\cdot,i]\in T$, $P\subset T$, $T[\cdot,i]$ 与 P 的连接强度定义为

$$\kappa(T[\cdot,i],P)=\frac{1}{\|P\|}\sum_{j=1}\frac{\sum_{k}^{m}h_{ki}h_{kj}}{\|H[\cdot,i]\|\,\|H[\cdot,j]\|}. \quad (5)$$

其中:$h_{ij}\in H_{m\times n}$ 表示 j 用户在一段时间内访问第 i 个 URL 的次数,$\|H[\cdot,i]\|$ 表示用户点击矩阵 $H_{m\times n}$ 的列向量 $H[\cdot,i]$ 的长度,$\|P\|$ 表示用户事务集 P 中用户事务的个数.

从以上定义可以看出,连接度表示类的内聚性. 聚类步骤如下:

1) 计算不同用户事务之间的 Manhattan 距离,对满足 $d_{ij}<\sigma$ 的用户事务进行合并,以此构成初始聚类;

2) 计算同一用户事务类中用户事务和类的连接度,对连接度大于指定阈值的类进行拆分,从而得到最终聚类.

算法描述如下:

Step1 用户事务聚类算法.

输入为用户事务集 $T=\{T[\cdot,i]\}$, $1\le i\le n$, σ 为聚类阈值. 输出为用户事务聚类. 功能是对用户事务进行聚类.

```
setup D_{n×n}^{M_d} by calcuting d_ij;
choose a σ as threshold, k: = 0;
while (T◇Φ) do
  K: = k + 1, C: = Φ;
  ∀T[·,i] ∈ T, let T = T - T[·,i],
  C = C + T[·,i];
  for (i: = 1; i ≤ ‖T‖; i + +) do
    using matrix D_{n×n}^{M_d};
    select all T[·,i](j) (d_ij < σ)
  {T[·,i](1), T[·,i](2), …,
  T[·,i](p)} from T, C_k
  where {T[·,i](1), T[·,i](2), …,
        T[·,i](p)} ∉ C;
  if {T[·,i](1), T[·,i](2), …,
  T[·,i](p)} ∈ T then
    T: = T - {T[·,i](1), T[·,i](2), …,
         T[·,i](p)};
    C: = C + {T[·,i](1), T[·,i](2), …,
         T[·,i](p)};
    end if
  end for
  C_k: = C;
```

end while;

$C:=\bigcup_k C_k$;

在此过程中,如果某一 T[·,i](j) 不在 T 中,则需要从已生成的类 C_k 中找到它,并加到 C 中,以处理类间的重叠问题.

Step2 聚类确认算法.

输入为初始聚类 $C=\{C_i\}$,ϕ为内聚阈值. 输出为最终用户事务聚类. 功能是对初始用户事务类进行确认.

for each $C_i \in C$ do
  for each T[·,j] $\in C_i$ do
    computing κ(T[·,j],C_i) according to (5)
    if κ(T[·,j],C_i) > ϕ then
      $C_i' = \cup\{T[\cdot,j]\}$;
      delete T[·,i] from C_i;
    end if
  end for
end for
append $\cup C_i'$ to C;

3.2 URL 聚类算法

如第 2 节所述,关联矩阵 $T_{m\times n}$ 的行向量 T[i,·] 反映了所有用户对本站点中不同 URL 的访问情况. 如果用户对某些 URL 的访问情况相同或相似,那么,这些 URL 应为相关的 URL,可以聚为一类.

行向量 T[k,·] 和 T[p,·] 的 Manhattan 距离 M_d' 计算如下:

$$M_d'(T[i,k],T[i,p])=\sum_{i=1}^{n}|t_{ik}-t_{ip}|. \quad (6)$$

聚类时,根据式 (6) 计算行向量之间的 Manhattan 距离,建立行向量间的距离矩阵 $D_{m\times m}^{M_d'}$,矩阵元素 $d_{ij}'(1\le i\le m,\ 1\le j\le m)$ 表示第 i 个 URL 和第 j 个 URL 之间的 Manhattan 距离.

对指定阈值 λ,若 $d_{ij}' \in D_{m\times m}^{M_d'}(1\le i\le m, i<j\le m)$,且 $d_{ij}'<\lambda$,那么就将第 i 个 URL 和所有的满足这个条件的第 j 个 URL 划分为一个类.

与 3.1 节一样,上述聚类过程仅考虑用户在某一 URL 的停留时间,而没有考虑用户对该 URL 的访问频率. 利用用户对 URL 的访问频率来对聚类的结果进行确认.

设 U 是 URL 的集合,任意 T[i,·] $\in U$,$G\subset U$,T[i,·] 和 G 的连接强度计算如下:

$$\kappa(T[i,\cdot],G)=\frac{1}{\|G\|}\sum_{j\in G}\frac{\sum_k^n h_{ik}h_{jk}}{\|H[i,\cdot]\|\ \|H[j,\cdot]\|}. \quad (7)$$

其中:$h_{ij}\in H_{m\times n}$ 表示 j 用户在一段时间内访问第 i 个 URL 的次数,$\|H[i,\cdot]\|$ 表示用户点击矩阵 $H_{m\times n}$ 的行向量 H[·,i] 的长度,$\|G\|$ 表示 URL 集 G 中 URL 的个数. 聚类同样分为两步:

Step1　计算不同 URL 事务之间的 Manhattan 距离,对满足 $d_{ij}'<\lambda$ 的用户事务进行合并,以此构成初始聚类;

Step2　根据式 (7) 计算同一 URL 事务类中 URL 事务和类的连接度,对连接度大于指定阈值的类进行拆分.

具体算法描述与 3.1 节类似,这里不再详述.

4　实验结果

利用 Microsoft Visual C++ 6.0 语言实现本文算法和文献[5,8] 的相关算法;然后在内存为 256 M,P Ⅳ 2.4 的 Windows XP 平台上进行相关性能测试与对比.

在美国加利福尼亚大学 Irvine 分校的 CIkdd 数据库仓库下载 URL 个数为 500 的 WWW 服务器日志文件,测试时以此为基础,提取其中的 50 000 条记录,形成 1 067 个用户事务;然后进行聚类.

首先测试算法的准确性. 为验证方便,测试时先设计 6 个不同的测试用例. 首先用手工的方法对这些数据进行分析,得到一定数目的相似用户事务群体和网站 URL;然后采用本文算法和文献[8] 中关于网站 URL 聚类算法以及文献[5] 中用户事务聚类算法进行聚类,聚类时适当地调整聚类阈值,如 σ 和 ϕ 的大小,得到与手工聚类数目相同的结果;最后将这些结果与手工得到的结果进行比较,得到图 2 所示的准确度曲线. 图 2 中 4 种算法从左至右依次为本文用户事务聚类算法,文献[5] 用户事务聚类算法,本文 URL 聚类算法,文献[8]URL 聚类算法. 图 2 表明,用户事务聚类的准确率在 79 % 左右,网站 URL 聚类的准确率在 76 % 左右,准确度高于文献[5,8] 的算法.

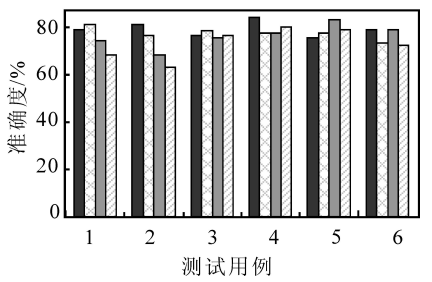

图 2　算法精度比较

然后测试算法的扩展性. 测试时将日志文件分割成大小为 100 k,300 k,500 k,700 k,900 k 和 1 100 k 的 6 个测试用例;然后记录采用本文算法和

文献[5,8]算法在上述 6 种情况下的 CPU 时间,得到如图 3 所示的曲线图. 对于大型网站而言,用户浏览矩阵和用户点击矩阵均为典型的稀疏矩阵,在算法具体实现时使用 3 元组对矩阵进行处理和存储. 而且,矩阵 $D_{n\times n}^{M_d}$ 是一次性计算并存储的,在算法的运行过程中不需要反复计算用户事务之间的相似度,因此算法的效率也有所提高.

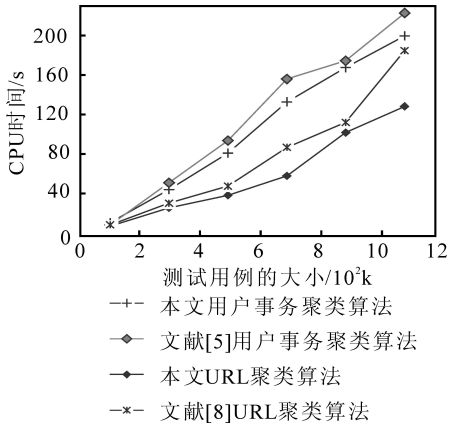

图 3 算法 CPU运行时间

聚类结果表明,对于用户事务和网站 URL 聚类,在任何一个测试用例里,本文算法的 CPU 时间均小于文献[5]用户事务和文献[8] 网站 URL 聚类算法的 CPU 时间,而且随着数据量的增多,本文算法 CPU 时间曲线的上升趋势比较缓慢,说明本文算法随着数据量的增加,运行时间增加较慢,具有较好的扩展性.

5 结 语

本文提出了一个以用户浏览离散化时间和用户浏览次数为度量的 Web 用户事务和网站 URL 聚类算法. 通过建立 Web 站点用户访问矩阵可以方便灵活地同时对 Web 用户事务和网站 URL 进行聚类. 在相似性度量上引入了浏览时间离散化的概念,并使用用户在网站 URL 上的浏览次数来定义连接强度对类进行确认. 实验表明,本文 Web 用户事务聚类和网站 URL 聚类算法与已有的相关算法相比,准确性高,运行时间短,扩展性好.

参考文献(References)

[1] Srivastava J, Cooley R, Deshpande M, et al. Web usage mining: Discovery and applications of usage patterns from web data[J]. SIGKDD Explorations, 2000, 1(2): 12-23.

[2] Cooley R, Mobasher B, Srivastava J. Data preparation for mining world wide web browsing patterns [J]. Knowledge and Information Systems, 1999, 1(1): 5-32.

[3] Mobasher B, Cooley R. Creating adaptive Web sites through Usage-based clustering of URLs[C]. Proc of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop. New York: IEEE Press, 1999: 32-37.

[4] Paliouras G, Papatheodorou C, Karkaletsis V, et al. Clustering the users of large web sites into communities [C]. Proc of the 17th Int Conf on Machine Learning. San Mateo: Morgan Kaufmann, 2000: 719-728..

[5] Fu Y, Sandhu K, Shih M. A generalization-based approach to clustering of Web usage session[C]. Web Usage Analysis and User Profiling. New York: Springer-Verlag, 2000: 21-38.

[6] Shahabi C, Zarski A M, Shah J. Knowledge discovery from users web-page navigation[C]. Proc of 7th Int Conf on Research Issues in Data Engineering. Birmingham: IEEE Computer Society Press, 1997: 20-29.

[7] 苏中, 马少平, 杨强, 等. 基于 Web-log mining 的 Web 文档聚类[J]. 软件学报, 2002, 3(1): 99-104.
(Su Z, Ma S P, Yang Q, et al. Document clustering based on Web-log mining [J]. J of Software, 2002, 3 (1): 99-104.)

[8] Perkowitz M, Etzioni O. Adaptive websites: Automatically synthesizing Web pages [C]. Proc of AAAI 98. Madison: AAAI Press, 1998: 35-40.

[9] Catledge L, Pitkow J. Characterizing browsing behaviors on the world wide Web [J]. Computer Networks and ISDN Systems. 1995: 27(6): 1065-1073.

[10] Cooley R, Mobasher B, Srivastava J. Grouping web page references into transactions for mining worldwide Web browsing patterns[C]. Proc Knowledge and Data Engineering Workshop. Newport Beach, CA: IEEE Press, 1997: 2-9.