文章编号:1001 - 9081(2009)04 - 1099 - 03

一种新的用户事务算法

高集荣[1],田 艳[2],邵海英[1]
(1. 中山大学 信息科学与技术学院,广州 510275; 2. 西安财经学院 信息学院,西安 710061)
(gaojr@mail.sysu.edu.cm)

摘 要:提出了双阈值用户事务算法。根据用户所访问的页面数来判断该用户是否为偶然用户,利用网络的拓扑结构和网页最低兴趣度来衡量一个网页是否为用户感兴趣的页面。改进了数据预处理过程,删除了偶然用户引起的访问记录,以及链接页面和用户不感兴趣的页面,生成一种有效的访问页面序列,即双阈值用户事务。通过事例对算法的有效性进行了论证。

关键词:数据挖掘算法;数据预处理;用户事务;双阈值
中图分类号: TP311.13 **文献标志码**:A

New user transaction algorithm

GAO Ji-rong[1], TIAN Yan[2], SHAO Hai-ying[1]
(1. *School of Information Science and Technology, Sun Yat-sen University, Guangzhou Guangdong* 510275, *China*;
2. *School of Information, Xi'an University of Finance and Economics, Xi'an Shaanxi* 710061, *China*)

Abstract: This study proposed a user business algorithm with double thresholds. This algorithm first acted according to the page number which the user visited to judge whether this user was the accidental user, and then the network topology and homepage lowest interest degree to judge whether the homepage appealed to the users. This method improved the data pretreatment process, and deleted the visit record which the accidental user caused, as well as the link pages and the pages that users were not interested in, produced one kind of effective visiting page sequence, namely double thresholds user business. This paper has proved the validity of the algorithm through an instance.

Key words: data mining algorithm; data preprocessing; user transaction; double thresholds

0 引言

数据预处理的结果作为日志挖掘算法的输入,直接影响到挖掘算法产生的规则与模式,影响着挖掘的质量,因此改进 Web 日志数据预处理技术可以有效地提高 Web 日志挖掘的质量。为此,人们对 Web 日志的预处理技术进行了广泛的研究[1-3],文献[4]提出了根据基于历史搜寻路径统计用户寻找目标花费的平均时间,用以量化 Web 页面的搜寻费用,提出了一种新的支持站点设计优化的 Web 使用挖掘方案。

当根据具体的分析要求进行数据挖掘时,我们可能仅仅需要考虑一些典型用户的使用模式,所以有时候就没有必要将偶然用户的记录也分析在内。对于这种噪音数据,我们必须在输入到聚类或者其他的分析步骤前就将其过滤掉。并且以上文献也没有考虑用户访问的页面并不一定是用户感兴趣的页面;用户用相同的时间访问不同的页面时,所表示的对页面的兴趣也不一定相同。

本文提出了双阈值用户事务预处理方法,改进后的 Web 预处理的过程分为数据清洗、识别用户、识别用户会话和双阈值事务四个阶段。

1 算法描述

该算法主要通过设置阈值来获得用户事务。首先通过设置一个阈值 a 来过滤掉偶然用户的访问记录,然后利用另外设置的一个阈值 b 来过滤掉非偶然用户的不感兴趣的页面。下面将对这两个阈值进行比较详细的描述。

1.1 偶然用户阈值 a

考虑一下在通常的用户访问过程中,我们不难发现,随着网站的增多,我们在一个网站上的驻留时间和浏览的网页可能是比较少的,而如果到达了一个我们比较感兴趣的网站后,驻留时间和浏览的网页数目会有很大的提高。对于一个具体的网站来说,在成千上万个用户中,有很大的一部分是仅仅浏览了一两个网页就离开了本网站,对于这种用户,我们显然不可能从中得到一个有用的、潜在的访问模式,因为我们可以看出他们对该网站的访问具有很大的偶然性,对具体网页的访问也有着很大的随机性,所以如果从这些用户的访问条目中提取用户的访问模式显然是不具有典型性和代表性的,而这种访问模式也就不能进一步地利用在网站的指导建设中和网页的推荐访问等后续的具体应用上。为此,我们需要过滤掉这些偶然用户引起的访问记录。

所谓的偶然用户是指对于一个网站而言浏览的网页数比较少,并且浏览时间也比较短的用户。

定义 1 偶然用户。偶然用户是指仅有一个用户会话,并且在该会话中,用户访问的网页数不超过阈值 a 的用户。通常阈值 a 有我们根据实际的网站情况决定。如定义 a 为网站总页面数的 1/10。

1.2 页面兴趣度阈值 b

改进网站结构,推荐用户感兴趣的页面,实现商业智能等,都应该基于用户所感兴趣的网页[5]。但是,在现实情况

收稿日期:2008 - 10 - 17;**修回日期**:2008 - 12 - 15。

作者简介:高集荣(1960 -),男,陕西西安人,副教授,硕士,主要研究方向:数据库技术、数据挖掘; 田艳(1962 -),女,四川成都人,教授,硕士,主要研究方向:数据库技术、网络技术及其应用; 邵海英(1985 -),女,山东济南人,硕士研究生,主要研究方向:数据库技术、数据挖掘。

下,用户所访问过的网页并不一定是用户感兴趣的网页。有时,用户为了到达自己喜欢的页面不得不访问某些链接网页。而这些链接页面显然不是用户感兴趣的。并且即便是用户访问过的内容页面也不一定就是用户感兴趣的。考虑这样一种情况:用户通过某个标题链接进入了一个内容页面,但却发现该页面介绍的内容并不是自己想要的,在这种情况下,用户可能会迅速离开该网页并继续寻找自己感兴趣的网页,显然这种内容页面也不是用户感兴趣的,尽管用户访问了该页面。而且即便是用户感兴趣的页面,用户感兴趣的程度也不一定是相同的,通常用户越感兴趣的页面,在其上花费的和时间也就越多。那么可不可以仅通过页面浏览时间来判断用户是不是对某个网站感兴趣呢? 答案是否定的。用户以相同的时间浏览两个网页时,并不表示用户对这两个网页同样的感兴趣。例如:如果用户浏览完 A 页面需要 10 min,浏览完 B 网页需要 2 min,当用户分别花 2 min 的时间来浏览两个网页时,显然用户对这两个页面的感兴趣程度是不一样的。针对上述问题,我们采用了兴趣度来确定用户对页面感兴趣的程度。下面给出兴趣度的定义。

定义 2 兴趣度。兴趣度 f 是指用户对于某个页面的感兴趣的程度。可以按如下方式计算:

1) 如果该页面不是最后一个页面,并且用户对该页面的访问时间大于该页面的最佳访问时间,则 $f = 1$;如果用户对该页面的访问时间小于该页面的最佳访问时间,则 f = 用户对该页面的实际访问时间 / 该页面的最佳访问时间。

2) 如果该页面是最后一个页面,则 f = 用户对前面页面的访问的兴趣度的平均值。

可以看出,兴趣度的计算与页面的最佳访问时间有关,那什么是网页的最佳访问时间呢? 所谓页面的最佳访问时间是指满足绝大多数用户在这段时间内,能够看完网页的所有信息的时间。这个和网站的信息量有关,本文采用页面内的字数与人的平均浏览速度的比值与其他信息的浏览时间的和作为网页的最佳访问时间。

从兴趣度的定义,我们还可以看出在兴趣度的计算中,我们对最后一个网页进行了单独处理,这有效地克服了会话识别时,统一将最后一个页面的访问时间看作会话识别的超时时间的缺点,避免了由于最后一个页面的兴趣度过高而带来的不精确性,更加客观地反映了用户的真实兴趣。

有了兴趣度,我们就可以比较客观地评价用户对某个页面的感兴趣程度。所谓用户感兴趣的页面,就是用户对该页面的兴趣度高于某个阈值 b 的页面,否则,就可以认为是用户不感兴趣的页面。

对于不同的网站,可以设定某一个阈值 b,以此来判断用户对页面感兴趣的程度。

2 算法实现

1) 定义非偶然用户至少浏览的网页数 a,兴趣度阈值 b。

2) 定义目标网站中的每个页面的数据结构 *page*。

3) 定义变量 i 表示当前正在处理的用户会话 $Session[i]$,初始值为 1。

4) 定义变量 *TotalF*,记录用户会话中除最后一个页面外的其余页面的兴趣度之和,初值为 0;定义变量 *pages* 记录用户会话中除最后一个页面外的其余页面的页面总数,初值为 $Session[i].Record[\,]$ 的长度。这两个变量用于计算最后一个页面的兴趣度 = *totalF/pages*。

5) 定义变量 $F[i][j]$ 表示用户对第 i 个会话中的第 j 个页面的兴趣度。

6) 程序描述,本算法中设定非偶然用户至少有一个用户会话且该会话中访问的页面数不少于 a。

```
while(i <= 用户会话总数)
{
  TotalF = 0;
  Pages = Session[i].Record[ ] 的长度;
  //如果该用户为偶然用户时,删除该用户会话
  if (Session[i].TotalNum == 1&&Session[i].Record[ ] 的长度 < a)
  {
    删除该用户会话;
  }
  else
  {
    //除去该会话中的非兴趣页面
    while(j < Sessionp[i].Record 的长度)
    {
      按兴趣度计算方法计算 F[i][j];
      //删除用户不感兴趣的访问页面记录
      if(Record[j] 是链接页面 ‖ F[i][j] < b)
      {
        删除该访问记录 Record[j];
        Pages --;
      }
      else
      {
        TotalF += F[i][j];
      }
    }
    if(最后一个访问页面是链接页面)
    {
      删除该页面;
      Pages --;
    }
    else
      计算最后一个页面的兴趣度: F[i][last] = TotalF/Pages;
  }
}
```

该算法完成了用户会话的清理,大大减小了挖掘的数据量。

3 实例描述及结果分析

取服务器上的一段日志测试上面的算法。假设此时的日志已经做过了数据清理工作,如表 1 所示。

表 1 中的数据识别出如表 2 所示的三个用户。其中,用户访问页面序列和时间中的表示用户对该页面的访问时间,如(A,29″)表示用户对 A 页面的访问时间为 29 s。

假设 timeout 设置为 30 min,那么根据上面的用户列表,我们可以得到四个用户会话,如表 3 所示。

网站的拓扑结构如图 1 所示。

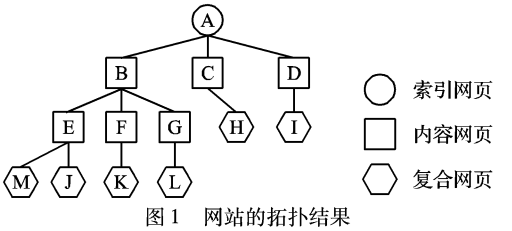

图 1 网站的拓扑结果

网页的分类为:

1) 索引页面:A;

2)复合页面:B、C、D、E、F、G;　　3)内容页面:H、I、M、J、K、L

表 1　用户清理后的数据

#	IP 地址	时间	方法 URL	代理
1	202.118.181.8	06-09-02 09:05:51	GET A.html	MSIE/6.0(Windows NT,5.1)
2	202.118.181.8	06-09-02 09:06:15	GET B.html	MSIE/6.0(Windows NT,5.1)
3	202.118.181.8	06-09-02 09:09:36	GET E.html	MSIE/6.0(Windows NT,5.1)
4	202.118.181.8	06-09-02 09:09:56	GET F.html	MSIE/6.0(Windows NT,5.1)
5	202.118.181.9	06-09-02 09:10:12	GET A.html	MSIE/6.0(Windows NT,5.1)
6	202.118.181.8	06-09-02 09:10:33	GET K.html	MSIE/6.0(Windows NT,5.1)
7	202.118.181.9	06-09-02 09:10:51	GET C.html	MSIE/6.0(Windows NT,5.1)
8	202.118.181.9	06-09-02 09:17:09	GET H.html	MSIE/6.0(Windows NT,5.1)
9	202.118.181.9	06-09-02 09:31:55	GET D.html	MSIE/6.0(Windows NT,5.1)
10	202.118.181.8	06-09-02 10:06:34	GET G.html	MSIE/6.0(Windows NT,5.1)
11	202.118.181.8	06-09-02 10:11:26	GET L.html	MSIE/6.0(Windows NT,5.1)
12	202.118.181.10	06-09-02 10:13:37	GET A.html	MSIE/6.0(Windows NT,5.1)
13	202.118.181.8	06-09-02 10:14:03	GET C.html	MSIE/6.0(Windows NT,5.1)
14	202.118.181.10	06-09-02 10:14:06	GET D.html	MSIE/6.0(Windows NT,5.1)

表 2　用户列表

用户 ID	用户 IP	用户 Agent	用户访问页面序列和时间
1	202.118.181.8	MSIE/6.0(Windows NT,5.1)	(A,24″),(B,3′21″),(E,20″),(F,37″),(K,56′01″),(G,4′52″),(L,2′37″),(C,30′)
2	202.118.181.9	MSIE/6.0(Windows NT,5.1)	(A,39″),(C,6′18″),(H,14′46″),(D,30′)
3	202.118.181.10	MSIE/6.0(Windows NT,5.1)	(A,29″),(D,30′)

表 3　会话列表

会话 ID	用户 ID	会话总数	页面访问序列和时间
1	1	2	(A,24″),(B,3′21″),(E,20″),(F,37″),(K,56′01″)
2	1	2	(G,4′52″),(L,2′37″),(C,30′)
3	2	1	(A,39″),(C,6′18″),(H,14′46″),(D,30′)
4	3	1	(A,29″),(D,30′)

假设针对该网站,非偶然用户最少应该访问的页面数为 3 个,该网站中的复合页面和内容页面的最佳访问时间分别(min)为:B(4)、E(3)、F(6)、K(10)、G(5)、L(10)、C(4)、H(12)、D(5),设页面的兴趣度阈值为 0.1。对于会话 1,由于 E 和 F 的兴趣度阈值小于 0.1,则表明用户对这两个页面并没有兴趣,删之。对于会话 3,尽管属于用户 2 的会话就它一个,但他所访问的页面数多于非偶然用户所访问的页面数,表明,该用户尽管这次可能访问的时间比较短,但由于访问的页面数比较多,也有可能是对该网站的某个信息感兴趣,比如,网上书店,用户可能需要浏览自己需要的新书是否到货,此时可能浏览时间很短,但却真实地代表了用户的兴趣。对于会话 4,由于对应的用户 3 仅有一个会话,且在该会话中,用户所浏览的网页数小于 3,所以删除该会话。经过处理后,得到如下的双阈值用户事务,如表 4 所示。

表 4　双阈值用户事务

会话 ID	用户 ID	会话总数	页面访问序列和时间
1	1	2	(B,0.8375),(K,0.8375)
2	1	2	(G,0.973),(L,0.262),(C,0.617)
3	2	1	(C,1),(H,1),(D,1)

若用普通的方法,识别的事务结果如表 5 所示。

该实例说明了双阈值用户事务算法与普通算法的不同。新方法识别的事务结构为 BK、GLC 和 CHD,以前的方法识别的结构为 BEFK、GLC、CHD 和 D。可以看出改进的方法识别的事务数目比较少,它删除了偶然用户引起的访问记录,并且相对于原来的事务,它的事务页面序列短,它删除了链接页面 A 和用户访问过的不感兴趣的页面,减少了数据量。

表 5　普通用户事务

会话 ID	用户 ID	会话总数	页面访问序列和时间
1	1	2	(B,3′21″),(E,20″),(F,37″),(K,56′01″)
2	1	2	(G,4′52″),(L,2′37″),(C,30′)
3	2	1	(C,6′18″),(H,14′46″),(D,30′)
4	3	1	(D,30′)

此外,如果存在多个用户同时访问了页面序列时,新算法能够很好识别这些用户是否具有相同的兴趣爱好。例如,用户 1 和用户 2 同时访问了 ABCD 四个页面,其访问时间分别为(单位为 min):用户 1:A(0.2)、B(2)、C(0.3)、D(2)和用户 2:A(0.2)、B(0.2)、C(5)、D(4)。按照以前的 Web 事务识别方法,则认为是相同的事务。但是,从用户的访问时间可以看出两个用户真正感兴趣的页面是不同的,用户 1 感兴趣的页面是 B 和 D,而用户 2 感兴趣的页面是 C 和 D,如果把这样的两个用户访问序列作为相同的事务,进行 Web 事务聚类,就很难发现真正的用户类。同样采用以前的事务识别方法识别出的结果(ABCD)进行页面聚类,同样也不能生成真正的用户感兴趣的页面组。

可以看出用本文的提出的双阈值用户事务识别方法识别 Web 事务,在减少用户事务数目的同时,还减少了页面序列的长度,从而减少了数据挖掘的数据量;同时还能区分相同的用户访问序列的不同访问事务,提高了挖掘的精确性。

(下转第 1105 页)

我们以下面的标准评价聚类结果:一个文本在聚类后仍属于原来所在类的下近似或上近似则认为该文本的聚类结果是正确的。表 2 给出了使用基于绝对距离的粗 K-means 方法得到的聚类结果,其中绝对距离阈值使用 $d = 0.05$,下近似和边界的权重 $\omega_{low} = 0.8$,$\omega_{up} = 0.2$,正确率为 92.5%;利用基于相对距离的粗 K-means 方法进行聚类,人工指定相对距离阈值 $\theta = 0.05$,下近似和边界的权重 $\omega_{low} = 0.8$,$\omega_{up} = 0.2$。在 253 篇文章中,只有 16 篇文章的聚类结果是错误的,正确率有所提高,为 93.6%。文本聚类结果的比较进一步说明了基于相对距离的粗 K-means 方法进行模糊聚类的有效性。

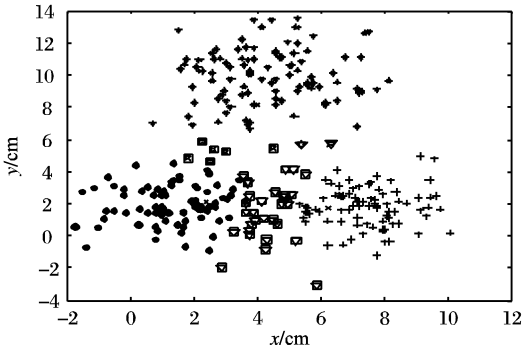

图 4　阈值较大时,基于绝对距离的聚类结果

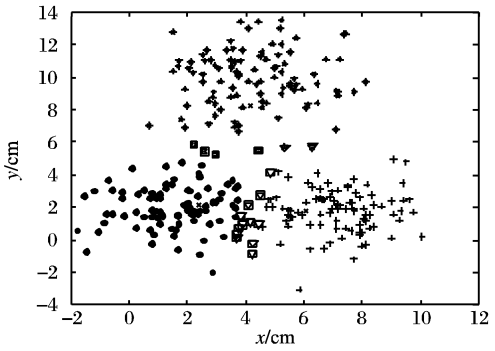

图 5　基于相对距离的聚类结果

表 2　利用基于绝对和相对距离的粗 K-means 方法得到的聚类结果比较

结果对比		文本聚类结果						
		只属于经济类	只属于政治类	只属于教育类	同时属于经济类和政治类	同时属于经济类和教育类	同时属于政治类和教育类	同时属于经济类、政治类和教育类
基于绝对距离分类结果	经济类(97)	41	4	1	33	13	2	2
	政治类(84)	2	37	1	28	2	11	2
	教育类(72)	4	0	24	3	26	14	1
基于相对距离分类结果	经济类(97)	56	3	0	23	11	1	2
	政治类(84)	2	48	1	17	2	10	3
	教育类(72)	3	1	32	2	21	12	1

5　结语

本文首先对现有的两种基于绝对距离的粗 K-means 方法进行了讨论,指出了各自的不足之处。在此基础之上,讨论了用相对距离替代绝对距离的合理性,从而给出了基于相对距离的粗 K-means 方法。通过对随机数据、Iris 数据和文本数据进行聚类效果比较,验证了基于相对距离的粗 K-means 方法的可行性和有效性。在基于相对距离的粗 K-means 方法中,距离阈值、权重参数的设置对聚类结果的影响是我们下一步要研究的问题。

参考文献:

[1] HAN JIA-WEI, KAMBER M. 数据挖掘概念与技术[M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2007: 17 - 19.

[2] KRISHNAPURAM R, JOSHI A, YI L. A fuzzy relative of the k-medoids algorithm with application to Web document and snippet clustering[C]// Proceedings of the 1999 IEEE International. Conference on Fuzzy Systems. Washington, D C: IEEE Computer Society, 1999: 1281 - 1286.

[3] 王明春, 王正欧. 基于粗集与遗传算法相结合的文本模糊聚类方法[J]. 电子与信息学报, 2005, 27(4): 548 - 551.

[4] 冯征. 一种基于粗糙集的 k-means 聚类算法[J]. 计算机工程与应用, 2006, 42(20): 141 - 142.

[5] SUSHMITA M, HAIDER B. Rough-fuzzy collaborative clustering [J]. IEEE Transactions on Systems, Man, and Cybernetics, 2006, 36(4): 795 - 806.

[6] LINGRAS P. Unsupervised rough set classification using genic algorithms[J]. Journal of Intelligent Information Systems, 2001, 16(3): 215 - 228.

(上接第 1101 页)

4　结语

本章针对数据预处理过程中存在问题,提出了一种新的事务识别方法:双阈值事务算法,并且对该算法进行了全面的介绍。首先本章介绍了算法提出的背景,然后对算法中的阈值进行了详细的说明,并对算法中用到的数据结构给出了明确的定义,在此基础上,给出了算法的伪代码实现,最后通过一个实例来进一步说明了该算法的工作过程以及优越性。

参考文献:

[1] STOICA I, MORRIS R, KARGER D, *et al.* Chord: A scalable peer-to-peer lookup service for internet applications[C]// Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2003: 149 - 160.

[2] 赵伟, 何丕廉, 陈霞, 等. Web 日志挖掘中的数据预处理技术研究[J]. 计算机应用, 2003, 23(3): 62 - 64.

[3] 孔昊, 周长胜. Web 日志挖掘预处理研究[J]. 北京机械工业学院学报: 综合版, 2005, 20(4): 28 - 30.

[4] 郭新涛, 梁敏, 阮备军, 等. 挖掘 Web 日志降低信息搜寻的时间费用[J]. 计算机研究与发展, 2004, 41(10): 1737 - 1745.

[5] LEONG B, LISKOV B, DEMAINE E D. EpiChord: Parallelizing the chord lookup algorithm with reactive routing state management [C]// Proceedings of the 12th International Conference on Networks: ICON 2004. Washington, D C: IEEE Computer Society, 2004: 270 - 276.