# 基于 URL 相似度的会话识别方法①

周松松, 马建红

(河北工业大学 计算机科学与软件学院, 天津 300401)

**摘 要**: 随着互联网的快速发展, Web 日志的用户行为分析已经成为互联网技术领域的研究热点之一. 会话识别是 Web 日志用户行为分析的关键步骤, 精准的会话识别是有效进行用户行为分析的基础. 本文在 IIS Web 日志分析的基础上, 提出了一种基于 URL 相似度的会话识别方法. 实验结果表明, 此方法可以有效的识别出用户的真实会话.

**关键词**: 数据预处理; 用户识别; 会话识别; 编辑距离; 最长公共子序列

## Session Identification Method Based on Similarity URL

ZHOU Song-Song, MA Jian-Hong

(School of Computer Science and Software, Hebei University of Technology, Tianjin 300401, China)

**Abstract**: With the rapid development of Internet, Web log user behavior analysis has become one of research hotspots in the field of Internet technology. Session identification is the key step in the Web log user behavior analysis. Accurate session identification is the foundation of effective user behavior analysis. IIS Web log analysis is presented in this paper on the basis of a session identification method based on known URL. The experimental results show that this method can effectively identify the user's real conversation.

**Key words**: data preprocessing; user identification; session identification; levenshtein distance; longest common subsequence

随着互联网的广泛应用, 电子商务、网络教育越来越普及, 个性化服务的概念也应运而生. 个性化服务是指通过分析用户的浏览行为和浏览内容, 对用户兴趣进行动态感知与预测, 主要涉及 Web 日志挖掘和文本挖掘技术[1].

Web 日志挖掘已成为数据挖掘技术中越来越受重视的领域之一, 挖掘中的预处理技术也变得非常重要. 据统计, 三分之二的数据挖掘分析家们认为在一个完整的数据挖掘过程中, 预处理在时间上占据整个日志挖掘过程的 60%以上. 数据预处理是 Web 日志中最基础和最频繁的核心工作, 预处理后的结果将直接影响到挖掘算法产生的规则和模式, 因此数据预处理过程在整个 Web 日志挖掘过程中占据着非常重要的地位, 是挖掘质量的保证. 数据预处理一般包括数据清理、用户识别、会话识别和路径补全等. 而会话识别的目的是将用户的所有访问序列分成多个单独的用户一次访问序列, 会话的真实性和精准度是衡量预处理质量的重要指标, 因此, 会话识别是 Web 日志挖掘的重要处理步骤之一[2].

国外学者较早地对 Web 日志挖掘中的预处理技术进行了研究, 取得了不错的研究成果, 同时国内也有大量学者对 Web 日志预处理的过程展开了研究.

李燕等[1]采用基于引用的会话识别算法来进行 Web 日志的会话识别. 周爱武等[3]通过对比常用的会话识别方法,提出了基于主页面的会话识别算法. 严奉华等[7]分别对 Timeout 方法、参引长度法进行改进, 提出了一种改进的会话识别方法.

通过实验对比, 本文所提出的基于 URL 相似度的会话识别方法, 相比基于主页面的会话识别算法和基于时间阀值的会话识别算法有较高的准确率.

① 收稿时间:2014-04-14;收到修改稿时间:2014-05-16

## 1 相关技术介绍

### 1.1 IIS Web 日志格式

以下详细的介绍 IIS Web 服务器所产生的日志的格式.

IIS 的 Web 日志包含 14 个字段, 分别是 date、time、s-ip、cs-method、cs-uri-stem、cs-uri-query、s-port、cs-username、c-ip、cs(User-Agent)、sc-status、sc-substatus、sc-win32-status 以及 time-taken. 具体的每个字段的解释如表 1:

表 1　IIS Web 日志的字段

| 字段 | 含义 |
|---|---|
| date | 日期 |
| time | 时间 |
| s-ip | 网站的 IP 地址 |
| cs-method | 请求的方式 |
| cs-uri-stem | 请求的 URL |
| cs-uri-query | 请求的参数 |
| s-port | 服务器端口 |
| cs-username | 用户名 |
| c-ip | 客户端的 IP |
| cs(User-Agent) | 用户代理 |
| sc-status | 协议状态 |
| sc-substatus | 协议子状态 |
| sc-win32-status | win32 状态 |
| time-taken | 请求所用的时间 |

### 1.2 基于主页面的会话识别方法

基于主页面的会话识别方法是以 Web 日志中用户访问页面序列的 URL 来作为划分会话的标准. 若一个用户访问的 URL 为站点的首页 URL, 则认为用户开始了一个新的会话. 该条 Web 日志记录就是用户新会话的开始访问记录, 而该记录之前的一条记录是用户上一次会话的最后访问记录. 另外, 用户所用的访问页面序列中的第一条记录是用户第一个会话开始的标志[3].

### 1.3 基于页面访问时间的启发式会话识别方法

定义 1. Web 访问日志记录集合(Web Access Log Set, *WALS*), *WALS* = <*date, time, s-ip, cs-method, …, time-taken*>, 其中各项的含义见表 1. WALS 按时间顺序记录在一段时间内用户访问网站的情况. 设 *L* 表示网站的所有 URL 集合, 那么 WALS 满足如下规则:

(1)WALS 属性 *cs-uri-stem* 的内容一定属于集合 *L*.

(2)*WALS* 中的记录一定是按时间顺序存放的.

定义 2. 用户集合(User Set, *US*), 设 *WALS* 中共有 *n* 条记录, *US* = <*UID*, <$URL_1$,$Date_1$>…<$URL_i$, $Date_i$>>, 其中, $1 \le i \le n$. *US* 满足如下规则:

⑴任意一个用户的访问记录都是按时间顺序存放的.

⑵*WALS* 中的记录都属于一个用户, *US* 中的记录也只能是 *WALS* 中的.

⑶*WALS* 中的每条记录只能属于一个用户.

基于页面访问时间的启发式会话识别方法首先要设定页面的访问时间阀值 $\eta$. 设 $US^k$=<$UID^k$,<$URL_i^k$,$Date_i^k$>…<$URL_j^k$,$Date_j^k$>>, 其中, $1 \le i \le n$、$US^k \in US$ 表示第 k 个用户集合; <$URL_j^k$,$Date_j^k$> ∈*WALS*,$1 \le j \le i$ 表示用户 k 的第 j 条记录. <$URL_j^k$,$Date_j^k$>和<$URL_{j+1}^k$,$Date_{j+1}^k$>是用户 k 的连续两条访问记录, 当 $Date_{j+1}^k - Date_j^k \le \eta$ 时, 我们认为这两条记录属于同一会话, 否则, 我们认为<$URL_{j+1}^k$,$Date_{j+1}^k$>是属于下一个会话的第一条访问记录, 一般 η 的取值为 10 分钟[4].

### 1.4 Levenshtein 算法

Levenshtein 距离, 又称编辑距离, 指的是两个字符串之间, 由一个转换成另一个所需的最少编辑操作次数. 许可的编辑操作包括将一个字符替换成另一个字符, 插入一个字符, 删除一个字符. 该算法的具体流程如下:

设有两个字符串 *strA*, *strB*. 它们的长度分别是 *strA.Length = m, strB.Length = n*.

⑴如果 *m=0*, 则最小编辑距离是 *n*; 若 *n=0*, 则最小编辑距离是 *m*.

⑵构造一个*(m+1)*(n+1)*的矩阵 *Array*, 并初始化矩阵的第一行和第一列分别是 *0-n,0-m*.

⑶两重循环, 遍历 *strA*, 在此基础上遍历 *strB*, 如果 *strA[i]=strB*, 那么 *cost=0*, 否则 *cost=1*. *Array[i]=min(Array[j-1][i]+1,Array[i-1]+1, Array[j-1][i-1]+cost),* 其中 *min* 表示取这三个数的最小值.

⑷循环结束后, 矩阵的最后一个元素就是最小编辑距离.

例如有两个字符串分别是 abc 和 abe. 如下图 1 所示.

| | abc | a | b | c |
|---|---|---|---|---|
| abe | 0 | 1 | 2 | 3 |
| a | 1 | A处 | | |
| b | 2 | | | |
| e | 3 | | | |

图 1 Levenshtein Algorithm

A 处的值取决于: 左边的 1、上边的 1、左上角的 0. 按照 Levenshtein 算法上面的值和左面的值都要求加 1, 这样得到 1+1=2. A 处由于是两个 a 相同, 所以的左上角的值加 0(否则加 1), 即 0+0=0. 这是有三个值, 左边的计算后为 2, 上边的计算后为 2, 左上角计算后的值为 0, 所以 A 处的值我们取这三处最小的 0. 以此类推我们得到最终的 Levenshtein Distance. 如下图 2 所示.

| | | a | b | c |
|---|---|---|---|---|
| | 0 | 1 | 2 | 3 |
| a | 1 | A处 0 | D处 1 | G处 2 |
| b | 2 | B处 1 | E处 0 | H处 1 |
| e | 3 | C处 2 | F处 1 | I处 1 |

图 2 Levenshtein Distance

依据上图我们的到这两个字符串的 Levenshtein Distance 为 1(I 处的值). 这时我们取两个字符串长度的最大值 *maxLength*, 用 1-(*LevenshteinDistance/maxLength*), 得到两个字符串的相似度. 例如 abc 和 abd 的相似度为 $\alpha$=1-(1/3)=0.666[5].

1.5 LCS

LCS 算法是求两个字符串的最长公共子串. 解法就是用一个矩阵来记录两个字符串中所有位置的两个字符之间的匹配情况, 若是匹配则为 1, 否则为 0. 然后求出对角线最长的 1 序列, 其对应的位置就是最长匹配子串的位置[6]. 设两个字符串中长度的最大值 *maxLength*, 公共字符串的长度为 *LCSDistance*, 这两个字符串的相似度为 $\beta$= *LCSDistance/maxLength*.

LCS 的性质: 记 $X_m = \{x_0,x_1,\ldots,x_{m-1}\}$, $Y_n = \{y_0,y_1,\ldots,y_{n-1}\}$ 为两个字符串, 并设 $Z_k = \{z_0,z_1,\ldots,z_{k-1}\}$ 是 *X* 和 *Y* 的一个最长公共子串. 则它满足下述几条性质:

(1)如果 $x_{m-1}=y_{n-1}$, 那么 $z_{k-1}=x_{m-1}=y_{n-1}$, 并且 $Z_{k-1}$ 是 $X_{m-1}$ 和 $Y_{n-1}$ 的一个 *LCS*.

(2)如果 $x_{m-1}\neq y_{n-1}$, 当 $z_{k-1}\neq x_{m-1}$ 时, *Z* 是 $X_{m-1}$ 和 *Y* 的 *LCS*.

(3)如果 $x_{m-1}\neq y_{n-1}$, 当 $z_{k-1}\neq y_{n-1}$ 时, *Z* 是 *X* 和 $Y_{n-1}$ 的 *LCS*. 该算法的具体流程如下:

设有两个字符串 *strA*, *strB*. 它们的长度分别是 *strA.Length* = *m*, *strB.Length* = *n*.

(1)根据输入的两个字符串 *strA*, *strB* 构建一个 *LCS* 矩阵 *LCSMatrix[m+1][n+1]*.

(2)两重循环, 遍历 *strA*, 在此基础上遍历 *strB*, 如果 *strA[i]=strB[i]*, 那么 *LCSMatrix[i]=1*. 否则, *LCSMatrix[i]=0*.

(3)循环结束后, 矩阵对角线上最长 1 的序列的和就是 *LCS* 的长度.

## 2 基于URL相似度的会话识别算法

基于页面访问时间的启发式会话识别算法最大的缺点就是不能识别长会话, 也就是说两条记录之间的时间间隔大于 10 分钟, 也有可能是一个会话. 而基于主页面的会话识别算法, 具有天然的不足, 不能单纯的依靠用户是否访问主页面作为会话识别的界限. 由于本文所采用的 IIS Web 日志的限制我们不能采用基于引用的会话识别算法来提高会话识别的准确率.

在大多数网站中, 考虑到信息的组织、查阅和检索的高效性, 大都是基于主题层级结构方式的 Web 目录组织网络. 因此, Web 目录可以用来描述一个网页的内容[2], 在此基础之上提出了基于 URL 相似度的会话识别算法.

定义 3. 用户会话集合(User Session Set, *USS*), 设 *WALS* 中共有 *n* 条记录, *USS*=<*USID*, <$URL_1$,$Date_1$>,…,< $URL_i$, $Date_i$ >>, 其中, $1\leq i\leq n$. *USS* 满足以下规则:

(1)任意一个用户会话中的访问记录都是按时间顺序存放的.

(2)*WALS* 中的记录都属于一个用户会话, *USS* 中的记录也只能是 *WALS* 中的.

(3)*WALS* 的每条记录只能属于一个用户会话.

定义 4. 设 Levenshtein 算法计算出的相似度为 $\alpha$, LCS 算法计算出的相似度为 $\beta$. 我们采用黄金分割比例来分配这两个相似度的权重. 令 $USS^k$=<$USID_k$,<$URL_1^k$,$Date_1^k$>…<$URL_i^k$,$Date_i^k$>>,其中, $1\leq i\leq n$ 、$USS^k\in USS$ 表示第 k 个用户会话集合; <$URL_j^k$,$Date_j^k$> $\in WALS, 1\leq j\leq i$ 表示用户会话 k 的第 j 条记录. 假设<$URL_j^k$,$Date_j^k$>和<$URL_{j+1}^k$,$Date_{j+1}^k$>是用户会话 k 的连续两条访问记录, 则 $\gamma$= *0.382* * $\alpha(URL_j^k, URL_{j+1}^k)$ + *0.618* * $\beta(URL_j^k, URL_{j+1}^k)$. 其中 $\gamma$ 为基于 URL

会话识别算法的相似度, 且满足以下条件:

⑴如果 $\gamma \le 0.2$, $<URL_{j+1}^{k}, Date_{j+1}^{k}> \in USS^{k}$;

⑵否则, $<URL_{j+1}^{k}, Date_{j+1}^{k}> \in USS^{k+1} (USS^{k} \ne USS^{k+1})$.

## 3 实验结果与分析

### 3.1 实验过程

#### 3.1.1 数据清理

数据清理的目的主要是为了删除 Web 日志中与挖掘算法无关的数据. 由于用户访问时一般不会显示地请求页面上的图形文件、CSS 等, 这些文件是根据 HTML 的超文本引用标记浏览器自动下载的. Web 日志挖掘的目的是获得用户的行为模式. 这些文件与挖掘的用户行为无关, 因此将 Web 日志中请求的URL后缀名为 jpg、css、ico、gif、png 等对应记录删除[2]. 数据清理的具体流程如图 3 所示.

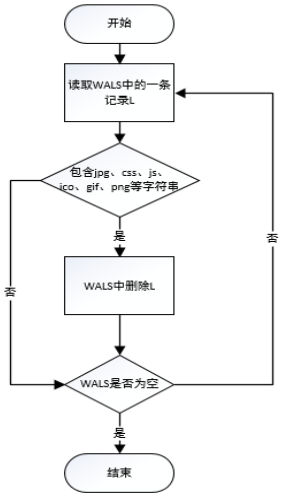



图 3　数据清理流程

数据清理后的日志文件的内容如下图 4 所示.

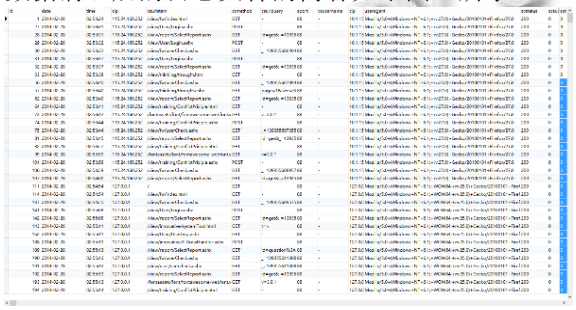


图 4　数据清理后的日志文件

#### 3.1.2 用户识别及会话识别

用户识别是将用户和请求页面相关联的过程, 它是会话识别的基础. 本文根据客户端 IP 地址(c-ip)、用户代理(User-Agent)来识别, 具体的规则如下:

⑴ 首先判断用户的 IP 地址, 不同的 IP 地址代表不同的用户;

⑵ 当 IP 地址相同时, 可以认为不同的操作系统或浏览器代表不同的用户. 用户识别的具体流程如图 5 所示.

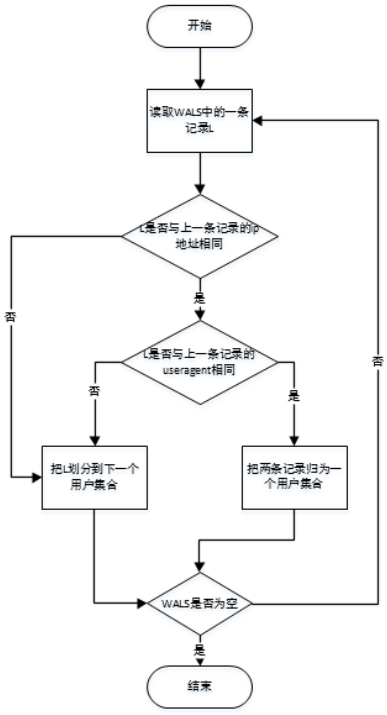
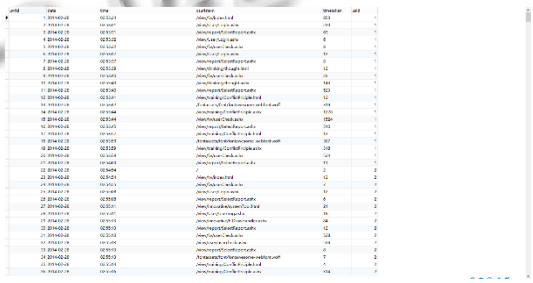



图 5　用户识别流程

经过用户会话识别后的日志文件如下图 6 所示.

图 6　用户识别后的日志文件



会话识别在用户会话识别的基础之上采用基于 URL 相似度的会话识别算法, 对按时间排列的用户会话日志记录逐一的进行识别. 会话识别的具体流程如图 7 所示.

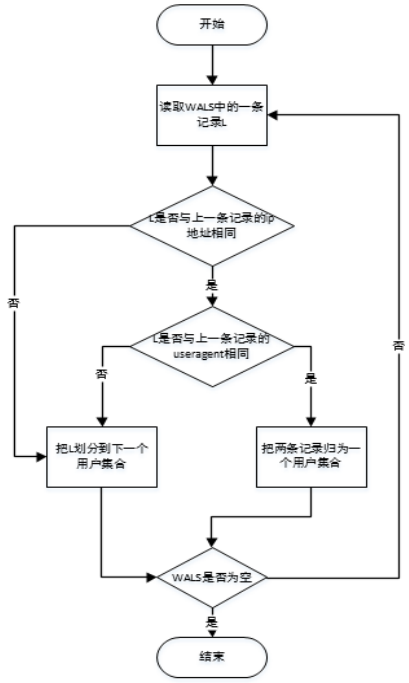



图 7 会话识别流程

经过本文所采用的会话识别算法得出的日志文件如下图 8.

| uvrid | date | time | csuristem | timetaken | ussid |
|---|---|---|---|---|---|
| 1 | 2014-02-28 | 02:53:24 | /view/fw/index.html | 853 | 1 |
| 2 | 2014-02-28 | 02:53:31 | /view/User/Login.ashx | 220 | 1 |
| 3 | 2014-02-28 | 02:53:31 | /view/report/SelectReport.ashx | 63 | 1 |
| 4 | 2014-02-28 | 02:53:32 | /view/User/Login.ashx | 6 | 1 |
| 5 | 2014-02-28 | 02:53:32 | /view/fw/userCheck.ashx | 8 | 1 |
| 6 | 2014-02-28 | 02:53:37 | /view/User/Login.ashx | 12 | 1 |
| 7 | 2014-02-28 | 02:53:37 | /view/report/SelectReport.ashx | 8 | 1 |
| 8 | 2014-02-28 | 02:53:39 | /view/thinking/thought.html | 12 | 1 |
| 9 | 2014-02-28 | 02:53:40 | /view/fw/userCheck.ashx | 23 | 1 |
| 10 | 2014-02-28 | 02:53:40 | /view/thinking/thought.ashx | 143 | 1 |
| 11 | 2014-02-28 | 02:53:40 | /view/report/SelectReport.ashx | 523 | 1 |
| 12 | 2014-02-28 | 02:53:41 | /view/training/ConflictPriciple.html | 12 | 1 |
| 13 | 2014-02-28 | 02:53:42 | /fontassets/font/fontawesome-webfont.woff | 319 | 1 |
| 14 | 2014-02-28 | 02:53:44 | /view/training/ConflictPriciple.ashx | 1278 | 1 |
| 15 | 2014-02-28 | 02:53:44 | /view/fw/userCheck.ashx | 1524 | 1 |
| 16 | 2014-02-28 | 02:53:45 | /view/report/SelectReport.ashx | 510 | 1 |
| 17 | 2014-02-28 | 02:53:57 | /view/training/ConflictPriciple.html | 12 | 1 |
| 18 | 2014-02-28 | 02:53:59 | /fontassets/font/fontawesome-webfont.woff | 307 | 1 |
| 19 | 2014-02-28 | 02:53:59 | /view/training/ConflictPriciple.ashx | 318 | 1 |
| 20 | 2014-02-28 | 02:53:59 | /view/fw/userCheck.ashx | 524 | 1 |
| 21 | 2014-02-28 | 02:54:00 | /view/report/SelectReport.ashx | 11 | 1 |
| 22 | 2014-02-28 | 02:54:54 | / | 2 | 1 |
| 23 | 2014-02-28 | 02:54:54 | /view/fw/index.html | 12 | 2 |
| 24 | 2014-02-28 | 02:54:55 | /view/fw/userCheck.ashx | 7 | 2 |
| 25 | 2014-02-28 | 02:55:08 | /view/User/Login.ashx | 12 | 2 |
| 26 | 2014-02-28 | 02:55:08 | /view/report/SelectReport.ashx | 6 | 2 |
| 27 | 2014-02-28 | 02:55:11 | /view/innovative/systemTool.html | 21 | 2 |
| 28 | 2014-02-28 | 02:55:11 | /view/User/UserImg.ashx | 18 | 2 |
| 29 | 2014-02-28 | 02:55:13 | /view/innovative/F_DrawHandler.ashx | 84 | 2 |
| 30 | 2014-02-28 | 02:55:13 | /view/report/SelectReport.ashx | 12 | 2 |
| 31 | 2014-02-28 | 02:55:13 | /view/fw/userCheck.ashx | 528 | 2 |
| 32 | 2014-02-28 | 02:55:13 | /view/user/usercheck.ashx | 534 | 2 |
| 33 | 2014-02-28 | 02:55:13 | /view/report/SelectReport.ashx | 8 | 2 |
| 34 | 2014-02-28 | 02:55:13 | /fontassets/font/fontawesome-webfont.woff | 7 | 2 |
| 35 | 2014-02-28 | 02:55:14 | /view/training/ConflictPriciple.html | 4 | 2 |
| 36 | 2014-02-28 | 02:55:16 | /view/training/ConflictPriciple.ashx | 314 | 2 |

图 8 基于 URL 相似度识别后的日志文件

### 3.2 结果与分析

本文的实验数据来源于河北工业大学计算机科学与软件学院 214 实验室所开发的 InventionKnowledgeCloud. 采用 2014 年 2 月 28 日到 2014 年 3 月 1 日的实验数据共 5332 条日志记录. 经过数据清理之后获得 2093 条有效的日志记录. 在此基础之上我们进行用户识别得到 12 个用户. 为了与其他会话识别方法进行会话识别, 我们分别用基于主页面和基于页面时间阀值的会话识别方法进行了会话识别, 其中基于主页面的会话识别方法识别出了 40 个用户会话, 而基于页面时间阀值的会话识别方法识别出了 13 个用户会话. 本文中提出的的基于 URL 相似度的会话识别方法识别出了 70 个用户会话, 可见本文的会话识别算法能够识别更多的会话. 为了验证会话识别算法的有效性我们对日志进行了人工的分析, 共得出了 112 个会话. 则基于主页面的会话识别算法的准确率为 35.7%, 基于页面时间阀值的会话识别算法的准确率为 11.6%, 本文所提到的会话识别算法的准确率为 62.5%.设 $\gamma$ 表示真实的会话个数, $\beta$ 表示通过会话识别算法识别出的会话个数, 则 $A_\pi=\frac{\beta}{\gamma}$, 表示会话识别算法 $\pi$ 的准确率. 已有的会话识别算法和加基于 URL 相似度的会话识别方法的实验结果如表 2, 这几种算法的优劣性如图 9.

表 2 实验结果

| 方法 | 会话个数 | | |
|---|---|---|---|
| | β | γ | $A_\pi$ |
| 时间阈值 | 13 | 112 | 11.6% |
| 主页面 | 40 | 112 | 35.7% |
| URL 相似度 | 70 | 112 | 62.5% |

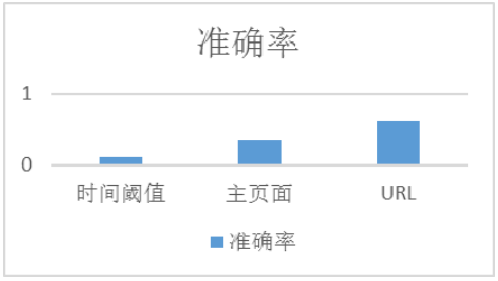



图 9 各种方法的准确率比较

进过实验比较本文所提出的基于 URL 相似度的会话识别算法相比其他两种会话识别算法能够识别更多的会话, 且准确率也较高.

## 4 结语

本文针对基于主页面和页面时间阀值的会话识别算法缺陷, 提出了基于 URL 相似度的会话识别算法, 实验结果显示, 此方法能有效的提高会话识别的精度, 使得会话识别出来的结果更接近与用户的真实会话. 在日志预处理的过程中可以明显的感觉出, 单节点对日志处理的局限性. 我们今后可以考虑对日志的预处理, 采用 MapReduce 框架对其进行分布式处理. 数据预处理的目的, 研究事务识别算法, 对用户会话进行分割和合并, 实现关联规则的发现也是下一步的研究方向.

## 参考文献

1 李燕,冯博琴,鲁晓锋.Web 日志挖掘中的数据预处理技术.计算机工程,2009,35(22):44-49.

2 张帅,陈兴蜀,童浩,崔晓靖.基于引用启发式和 URL 语义相结合的会话识别方法.计算机应用研究,2013.http://www.cnki.net/kcms/detail/51.1196.TP.20130809.1753.038.html.

3 周爱武,程博,李松长,夏松.Web 日志挖掘中的会话识别方法.计算机工程与设计,2010,31(5):936-964.

4 Spiliopoulou M, Mobasher B, Berendt B, et al. A framework for the evaluation of session reconstruction heuristics in web usageanalysis. INFORMS Journal on Computing, 2003, 15(2): 171-172.

5 计算字符串的相似度算法. http://wdhdmx.iteye.com/blog/1343856

6 LCS 两个字符串最长公共子串.http://www.doc88.com/p-789750010537.html

7 严奉华,刘建平,杨凡丁.改进的 Web 访问日志会话识别算法.计算机工程与设计,2008,29(22):5685-5687.