基于缩进轮廓的HTML文档重复模式挖掘方法

朱沿旭[1] 王怀民[1] 史殿习[1] 尹 刚[1] 袁 霖[1,2] 李 翔[1]
(国防科学技术大学计算机学院 长沙 410073)[1] (信息工程大学电子技术学院 郑州 450004)[2]

摘 要 HTML文档重复模式挖掘是找到Web页面编码模版的关键,是Web数据自动抽取和Web内容挖掘的基础。传统的基于字符串匹配和树匹配的重复模式挖掘方法虽然具有较高的精确度,但是其性能对于处理海量的Web页面来说仍然是一个挑战。为了提高性能,提出了一种基于缩进轮廓的HTML文档重复模式挖掘方法。该方法首先定义了缩进轮廓模型,是一种由HTML文档每行代码的缩进值及行首的HTML标签构成的数据结构,它是HTML文档的一种简化抽象;该方法通过检测缩进轮廓中的串联重复波段,间接地挖掘HTML文档中的重复模式。实验表明,该方法不但具有较高的精确度,而且较明显地提升了性能。

关键词 重复模式挖掘,Web数据抽取,Web内容挖掘,缩进轮廓,串联重复波段
中图法分类号 TP391 **文献标识码** A

Indent Shape Based Approach for Mining Repeated Patterns of HTML Documents

ZHU Yan-xu[1] WANG Huai-min[1] SHI Dian-xi[1] YIN Gang[1] YUAN Lin[1,2] LI Xiang[1]
(School of Computer, National University of Defense Technology, Changsha 410073, China)[1]
(School of Electronic Technology, Information Engineering University, Zhengzhou 450004, China)[2]

Abstract Mining repeated patterns is the key to find encoding templates of Web pages, which is the basis for automatic Web data extraction and Web content mining. Existing approaches such as tree matching and string matching can detect repeated patterns with high precision, but their performance is still a challenge for massive Web pages processing. In order to improve performance, the paper presented a novel indent shape based approach for mining repeated patterns of HTML documents. Firstly, the approach defines the indent shape model, which is a kind of simplified abstraction of HTML documents consisting of indents and first tags of each line; Then, it detects repeated patterns indirectly by identifying tandem repeated waves from indent shape. Extensive experiments show that our approach achieves better performance compared with existing approaches.

Keywords Mining repeated patterns, Web data extraction, Web content mining, Indent shape, Tandem repeated waves

1 引言

Web数据是一种半结构化的数据,是后台数据库的数据记录按照一定的页面模版(Templates)由计算机程序展现在网页上的[1]。模版是HTML所表现出来的某种规则结构,使用相同模版生成的Web数据记录在结构上就表现出一种重复特性,重复模式就是这种重复特性的形式化表述。如图1所示,列表(a)的HTML源码为(b),其中加粗部分具有明显的重复特性,该列表模版的重复模式及数据抽取结果如表1所列。

微软亚洲研究院曾经从互联网中收集了近3亿个网页,从中获取了近19.5亿个列表,平均每一个网页中包含6.5个列表,这些列表包含了各种对象的重要信息[2],这充分表明Web列表是Web数据的重要载体之一。本文着重讨论对于列表重复模式的挖掘,主要任务就是从各种页面中识别列表模版,并将其转化为抽取规则,抽取Web对象的数据并存储在本地。重复模式挖掘是数据抽取的基础,通过它可以获取和整合来自多个网站和网页的数据,以提供很多增值服务,比如Web信息集成、比较购物、垂直搜索等。

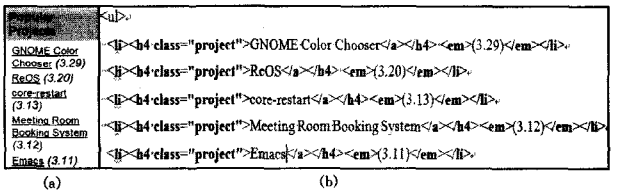

图1 列表模版

到稿日期:2010-09-29 返修日期:2010-12-28 本文受国家863计划重点课题(2007AA010301),国家自然科学基金(60903043),核高基重大专项课题(2009ZX01043-001)资助。

朱沿旭(1982—),男,博士生,CCF学生会员,主要研究领域为Web数据挖掘、社会网络,E-mail:zhu.yannick@gmail.com;王怀民(1962—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为分布式计算、可信软件、网络与信息安全;史殿习(1966—),男,博士,副教授,主要研究领域为分布计算与自主计算;尹 刚(1975—),男,博士,讲师,主要研究领域为可信软件、分布计算与信息安全;袁 霖(1981—),男,博士生,讲师,主要研究领域为软件可信评估和数据挖掘;李 翔(1988—),男,硕士生,主要研究领域为数据挖掘技术。

表1 列表重复模式及抽取结果

重复模式：〈li〉〈h4 class="project"〉name〈/a〉〈/h4〉〈em〉(number)〈/em〉〈/li〉	
GNOME Color Chooser	3.29
ReOS	3.20
Core-restart	3.13
Meeting Room Booking System	3.12
Emacs	3.11

传统的重复模式挖掘方法[3-6]主要有两种：字符串匹配[4]和树匹配[5]。两种方法的共同点是：将HTML文档抽象为编码字符串或者DOM树，然后通过遍历子串或者子树的方式，查找字符串中重复出现的片段或者DOM树中重复出现的子树。由于两种方法都使用了遍历，因此算法复杂度是$O(N^2)$，对于字符串匹配的方法来说，N代表字符数；对于树匹配的方法来说，N代表DOM树中的标签结点的个数。当处理海量页面的时候，这种复杂度带来的时间开销是不可接受的。

近年来研究者提出很多启发式的方法，目的是减少传统方法的遍历次数，最具代表性的方法主要有基于预定义标签[7,8]和基于视觉信息的启发式方法[9-12]。第一种方法利用HTML规范中对列表和表格标签的定义作为启发规则，比如〈table〉和〈ul〉分别代表列表的开始，〈tr〉和〈li〉分别代表列表表项的开始。这种方法虽然可以获得较高的性能，但是不能处理不是由预定义的标签组织的表格。第二种方法利用Web页面的视觉信息作为启发规则，比如文献[12]通过视觉模块之间的分隔符来识别列表和表项的边界。这种方法由于需要访问浏览器渲染引擎才能获得视觉模块的信息，因此引入了更多的额外开销。

本文通过观察HTML文档的标准缩进格式，发现列表对应的代码段左侧轮廓表现出了明显的有规律的凹凸特性，很好地表征了列表模版的重复特性。基于这个观察，文章首先提出了缩进轮廓模型，模型保持了列表模版的重复特性，同时简化了HTML文档的表示；然后，文章提出了基于HTML文档缩进轮廓的Web列表挖掘方法；最后，实验表明该方法的性能较之传统方法有大幅度提升。

2 缩进轮廓模型

HTML文档通常使用标签组织成嵌套结构，每一层嵌套代表一个HTML元素。每一个HTML元素都由3个部分组成[13]：一个开始标签、内容和一个结束标签，代码表示为〈tag〉element content〈/tag〉，在两个标签之间，可以嵌套包含底层的子元素。根据这个基本的HTML代码标准，本文给出以下假设，我们称其为标准HTML缩进格式：(1)每一个HTML元素的开始标签和结束标签都占用单独的一行；(2)底层子元素的标签对的位置相对于上层直接父元素增加一个缩进单元(比如两个空白符)；(3)同一个元素的开始标签和结束标签具有相同的缩进。

当然，不是每一个HTML文档都严格遵守上述的标准缩进风格。比如查看www.google.com.hk页面的HTML源码，可以发现它根本没有换行，仅仅具有一行连续的HTML代码；还有一些极端的情况，每一行代码都没有缩进，都是紧靠文档左侧边线。幸运的是，已经有很多HTML标准化工具比如Tidy(http://www.w3.org/People/Raggett/tidy)，可以将HTML转化成标准格式，转化算法的复杂度是$O(N)$，其中N是文档标签总数。另外，根据我们初步实验表明，在5000个HTML文档中，有超过94.5%的文档遵循标准格式。

HTML文档中的每一行开始的空白字符的个数，代表了这一行的缩进值，本文称其为缩进距离(indent distance)。每一个元素的开始标签和结束标签，也就是每一行的第一个标签称为框架标签(skeleton tag)，他们是组成缩进轮廓的基础。我们将每一行的行号和这一行的缩进值投影到二维坐标系中，可以得到一个坐标点，本文称其为标签点(tag point)。有了这些基本的概念，下面给出缩进轮廓的定义。

定义1(缩进轮廓) 给定一个n行的HTML文档，其缩进轮廓是一个n维的向量$\langle(p_1,t_1),(p_2,t_2),\cdots,(p_n,t_n)\rangle$，其中$t_i$是第$i$行的框架标签，$p_i$第$i$行的标签点；$p_i$表示成一个二元组$(i,id_i)$，$id_i$是第$i$行的缩进距离$(1\leqslant i\leqslant n)$。

我们将n个标签点用直线连接起来，会得到一条之字形的形状，称其为缩进轮廓线，其中任意两个标签点之间的部分定义为一个片段(segment)。假设一条缩进轮廓中，存在一个片段w，我们使用$SLN(w)=i$和$ELN(w)=j$来表示这个片段的起始和终止位置。如果存在两个片段w和w'，满足$SLN(w')\leqslant SLN(w)$并且$ELN(w')\geqslant ELN(w)$，那么w是w'的子段，记为$w\subseteq w'$；如果w和w'满足$SLN(w')=ELN(w)+1$或者$SLN(w)=ELN(w')+1$，则w和w'是连续的。

本文使用经典的计算方法来计算两个子段的相似度，比如Jaccard's相关系数、余弦定理和Dice's相关系数等。为了简单起见，选择第3种加以介绍。给定相同片段w中的两个子段w_i和w_j，取它们对应的标签向量$st=\langle t_1,t_2,\cdots,t_k\rangle$和$st'=\langle t_1',t_2',\cdots,t_k'\rangle$，向量是由片段中不重复的框架标签构成。相似度计算公式如下，

$$Dice(st,st')=2|st\cap st'|/(|st|+|st'|) \quad (1)$$

如果相似度大于等于给定的阈值T，那么w_i和w_j相似；如果相似度等于1，那么w_i和w_j完全相同。如果w中所有相邻片段相似度的平均值大于等于阈值T，那么w是一个串联重复波段。这个相似度的平均值被称为w的自相似度(self-similarity)。

定义2(串联重复波段) 给定一个缩进轮廓α和一个片段w，如果存在k个子片段$w_i(i=1,2,\cdots,k)$，$w=w_1w_2\cdots w_k$，满足以下两个条件：(i)w_i与w_{i+1}相似或者w_i与w_{i+1}完全相同；(ii)w_i和w_{i+1}是连续的，那么w是一个串联重复波段。

定义3(最大串联重复波段) 给定一个串联重复波段w，w是一个最大串联重复波段，当且仅当满足以下条件：对于每一个串联重复波段w'，如果$w\subseteq w'$则$FLN(w)=FLN(w')$并且$LLN(w)=LLN(w')$。

根据以上的基本定义，具有重复模式的列表对应的缩进轮廓片段就是最大串联重复波段，其子片段对应列表中的表项；如果列表包含嵌套列表，那么嵌套列表对应的缩进轮廓片段是串联重复波段，并且包含于父列表对应的片段。列表重复模式挖掘的问题就可以转化为查找给定缩进轮廓中的所有最大串联重复波段。

3 基于缩进轮廓的重复模式挖掘

基于缩进轮廓方法的关键是，如何将串联重复波段从缩

进轮廓中分离出来。文章提出了波段分解的方法，方法主要有4个步骤：(1)使用缩进直线水平扫描缩进轮廓，将缩进轮廓切割分段；(2)计算每一个片段的自相似度；(3)通过自相似度识别所有串联重复波段；(4)通过分析串联重复波段之间的包含关系，找到最大串联重复波段。

给定一个HTML文档 α 和一个阈值 T，α 对应的缩进轮廓表示为 λ。图2展示了方法的整个流程，其中indent line表示缩进直线，即直线 $y=\mathrm{id}$(id是 λ 中任意一个缩进距离)；SP(λ 中最小的缩进距离)和EP(λ 中最大的缩进距离)分别表示缩进直线的起始和终止位置，它们限定了缩进直线扫描的边界；黑色圆圈标注了缩进直线与缩进轮廓的交点intersection；右侧的箭头表示位于缩进直线上方、下方的位置关系。方法使用缩进直线沿着 y 轴的方向在SP和EP之间扫描 λ("扫描"代表id遍历 λ 中所有不重复的缩进距离)，第一步，缩进直线和 λ 的交点将轮廓分为很多片段，所有位于缩进直线上方的连续片段被合并为一个单一的片段，作为候选片段；然后，如果用来合并候选片段的连续片段的个数大于等于2，则计算该候选片段的自相似度，这里主要是假设一个列表中至少要包含两个或以上的表项；第三步，如果自相似度大于或者等于 T，则这个片段为一个串联重复片段，文章实验部分会详细介绍阈值 T 的选定方法；最后，检查所有串联重复片段之间的包含关系，没有包含于其他任何片段的候选片段就是最大串联重复波段。方法的伪代码如图3所示。

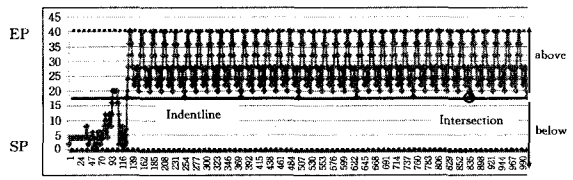

图2 方法流程

```
ISLM (Indent shape λ, Threshold T)
Output: set of greatest tandem repeated waves
1 for each distinct indent distance occurred in λ
2   get the intersections of the indent line and λ
3     merge contiguous segments above or on the indent line into one
      segment, which inserted into set S
4     for each segment Si of S
5       if the number of sub-segments contained by Si≥2
6         calculate self-similarity of Si
7         if self-similarity of Si≥ T
8           insert Si into set of tandem repeated waves TRW
9 for each element of TRW
10   if it is contained by any other element
11     delete it from set
12 return TRW
```

图3 方法伪代码

方法复杂度分析如下：该方法有3个循环，分别出现在第1,4,9行。第1行是遍历 λ 中 k 个不重复的缩进距离。第4行是遍历所有候选片段，这里的开销主要由自相似度计算产生，由于只需要计算相邻子片段之间的相似度，因此每一个子片段最多与其左右两个相邻子片段进行比较。假设 N_1 表示给定缩进轮廓(中不重复的(distinct)框架标签的个数，那么第4－8行的计算开销为 $O(2N_1)$；由于第4行的循环嵌套于第1行循环中，因此方法执行到第8行时的开销为 $O(2kN_1)$；第9行是遍历所有串联重复波段，λ 中的串联重复波段的个数远小于 N_1，同时计算串联重复波段包含关系的时候仅仅是比较波段的起始和终止位置，所以第9－11行的开销可以忽略不计。另外，之前提到的HTML文档缩进格式标准化算法开销为 $O(N)$，N 是HTML标签总个数；本文方法的总开销为 $O(2kN_1+N)$，在最坏的情况下HTML文档中的每一个标签都是不重复的，即 $N_1=N$，此时方法的复杂度可以表示为 $O((2k+1)N)$。

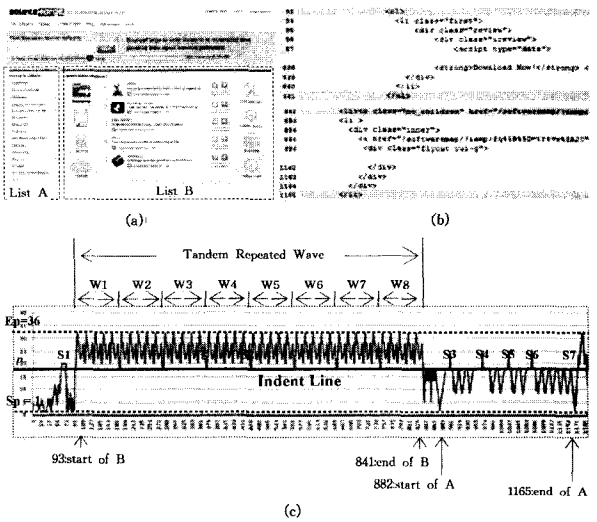

图4 案例分析

为了更好地解释我们的方法，图4给出一个案例分析。图4(a)展示了某网页(www.sourceforge.net)的页面快照，该页面包含两个列表 A 和 B(该网页实际上包含不止两个列表，这里为了简化描述，仅仅介绍其中的两个)；图4(b)展示了列表 A 和 B 对应的HTML代码片段；图4(c)展示了网页对应的缩进轮廓线，其中 $SP=1$，$EP=36$。当缩进直线移动到 $p=8$ 时，会发现列表 A 对应的串联重复波段，其边界为(882,1165)；当缩进直线移动到位置 $p=18$ 时，会得到7个候选片段 S_1-S_7，其中 S_2 包含8个连续的子片段 w_1-w_8，经过自相似度计算，S_2 为串联重复波段，对应列表B，其边界为(93,841)；当缩进直线移动到 $p=20$ 时，会发现 w_1-w_8 也都是串联重复波段。最终，方法会标识出两个最大串联重复波段。

4 实验及结果分析

本文的实验主要采用两个数据集，一个是文献[5]提供的数据集1，另一个是作者自己构造的数据集2。数据集1是从不同领域的22个网站中(新闻、图书等领域)随机挑选的具有相似模版的网页，每个网站平均挑选5个网页，平均每个网页中包含45个列表，列表种类多；数据集2主要是从8个知名的开源软件社区中随机挑选的网页，平均每个网页包含13个列表，列表种类较单一，主要是展示了该社区收录的部分项目列表。数据集的真实值是通过人工观察HTML源代码而得到的，标注了每个网页中包含的列表及其边界。实验将本文方法自动挖掘的列表和真实值进行比对，通过精确度、召回率和F-Score 3种评价指标来衡量方法的性能。精确度、召回率和F-Score的计算公式如(2)、(3)、(4)，其中TP代表被算法正确标识的列表个数，FP代表被算法误判的列表个数，FN代表没有被算法发现的列表个数。本文同时实现了MDR[5]和

IEPAD[4]的方法，并与本文的方法进行比较，如表2和表3所列。

$$Precision = TP/(TP+FP) \quad (2)$$

$$Recall = TP/(TP+FN) \quad (3)$$

$$F\text{-}score = 2 * Precision * Recall/(Precision+Recall) \quad (4)$$

3种方法都在同样的机器上进行测试，机器配置为SONY VGN-NW18H，Intel Core CPU T6500 2.1GHZ * 2，4GB of memory，32bit OS Win7。IEPAD和MDR的阈值T按照其作者的建议选定为0.5和0.3。如图5所示，本文通过设定不同阈值计算方法各项指标，发现当阈值设定为0.5时精确度、召回率同时达到最高值96.7%，所以，本文方法的阈值选定为0.5。

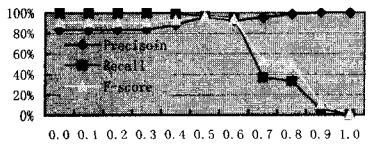

图5 方法性能随阈值变化曲线

表2 方法精确度测试结果对比

Data Set	Data Set 1			Data Set 2		
Approach	IEPAD	MDR	ISLM	IEPAD	MDR	ISLM
Precision	67%	98.4%	98%	79%	96.4%	96.7%
Recall	39%	99%	100%	61%	96%	96.7%
F-score	0.49	0.99	0.98	0.69	0.96	0.967

表3 方法执行时间对比

Page	Total characters	Total tags	Execute Time(ms)		
			IEPAD	MDR	ISLM
sourceforge.net	67624	3975	629154	2918	489
gitorious.org	11053	857	1433	109	9
appache.org	96876	4732	273	843	175
savannah.gnu.org	17358	832	400	48	21
gnome.org	17411	1206	1837	63	38
codeplex.search	105630	4624	919594	813	182
gna.project	15505	781	253	47	17
origo.project_list	23862	921	327	30	3
Average	44414.88	2241	194158.9	608.875	116.75

表2的结果显示：(1)通过两个数据集的测试，本文的方法在3种性能指标上要远优于IEPAD；(2)对于数据集1来说，本方法的召回率可以达到100%，其精确度和其他指标与MDR不相上下；(3)对于数据集2来说，本方法各项指标均优于MDR和IEPAD。表3的结果显示：本方法是3种方法中执行时间最短的，其平均执行时间比MDR快5倍、比IEPAD快1600倍。

算法精确度在两个数据集上所表现出来的差异主要由以下3个原因造成：(1)数据集1中的列表种类多表项少，表项少的列表对应的缩进轮廓线的重复特征不明显；(2)数据集1中的部分列表包含多层嵌套结构，这种嵌套结构较为复杂，容易造成父表的漏判；(3)很多由用户评论构成的列表结构较自由[14]，表项之间的结构相似度低。而数据集2中的列表种类少、表项多，列表结构多为平坦结构，重复特征很明显。实验结果说明本文的方法对于领域内的列表挖掘更加高效。本文将来的工作主要针对上述3个原因，进一步改进算法以提高其性能。

结束语 本文通过观察HTML代码缩进格式与网页模版重复模式之间的对应关系，提出了一种基于HTML文档缩进轮廓的重复模式挖掘方法。该方法首先定义了缩进轮廓模型，相比传统的字符串和DOM树简化了HTML文档的表示；其次，在缩进轮廓模型的基础上，提出了缩进轮廓线波段分解的方法，通过缩进直线将缩进轮廓线切割分段，并计算每一个片段的自相似度，将自相似度大于等于阈值T的串联重复波段从轮廓中标识出来。本方法将传统的重复模式挖掘问题转化为缩进轮廓串联重复波段挖掘问题。最后，实验表明本方法在精确度及执行时间等方面都优于传统方法。

参考文献

[1] Liu Bing. Exploring Hyperlinks, Contents, and Usage Data[M]. Berlin Heidelberg New York: Springer, 2007: 323-379

[2] Wang Jing-jing, Shao Bin, Wang Hai-xun, et al. Understanding Tables on the Web[R]. Microsoft Research Asia, 2011

[3] Embley D W, Jiang Y S, Ng Y K. Record-Boundary Discovery in Web Documents[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Philadelphia, PA, 1999: 467-478

[4] Chang C-H, Lui S. IEPAD: Information Extraction Based on Pattern Discovery[C]// Proceedings of the International World Wide Web Conference. Hong Kong, China, 2001: 681-688

[5] Liu Bing, Grossman R, Zhai Yan-hong. Mining Data Records in Web Pages[C]// Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, DC, USA, 2003: 601-606

[6] Jindal N, Liu Bing. A Generalized Tree Matching Algorithm Considering Nested Lists for Web Data Extraction[C]// Proceedings of the SIAM International Conference on Data Mining. Columbus, Ohio, USA, 2010: 930-941

[7] Lin S-H, Ho Jan-ming. Discovering Informative Content Blocks from Web Documents[C]// Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Alberta, Canada, 2002: 588-593

[8] Diao Yan-lei, Lu Hong-jun, Chen Song-ting, et al. Toward Learning Based Web Query Processing[C]// Proceedings of the International Conference on Very Large Data Bases. Cairo, Egypt, 2000: 317-328

[9] Simon K, Lausen G. ViPER: Augmenting Automatic Information Extraction with Visual Perceptions[C]// Proceedings of the ACM CIKM International Conference on Information and Knowledge Management. Bremen, Germany, 2005: 381-388

[10] Gatterbauer W, Bohunsky P, Herzog M, et al. Towards Domain Independent Information Extraction from Web Tables[C]// Proceedings of the International World Wide Web Conference. Banff, Alberta, Canada, 2007: 71-80

[11] Liu Wei, Meng Xiao-feng, Meng Wei-yi. ViDE: A Vision-Based Approach for Deep Web Data Extraction[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 22(3): 447-460

[12] Zhao Hong-kun, Meng Wei-yi, Wu Zonghuan, et al. Fully Automatic Wrapper Generation for Search Engines[C]// Proceedings of the international conference on World Wide Web. Chiba, Japan, 2005: 66-75

[13] W3C. HTML 4.01 Specification [S]. http://www.w3.org/TR/html401: 1999

[14] Song Xin-ying, Liu Jing, Cao Yun-bo, et al. Automatic extraction of web data records containing user-generated content [C]// Proceedings of the ACM Conference on Information and Knowledge Management. Toronto, Ontario, Canada, 2010: 39-48