# 一种面向 WEB 页面的标记聚类方法*

焦永强[1]　王维扬[2]　尚　颖[2]
（1. 中国航空综合技术研究所　北京　100028）（2. 北京化工大学　北京　100029）

**摘　要**　针对Web测试中现有Web页面聚类方法无法准确描述复杂页面结构、页面聚类准确度低、时间复杂度高的问题，分析了Web页面的DOM结构和节点属性，给出改进的树匹配算法衡量Web页面间相似度，并提出一种新的标记聚类方法实现Web页面聚类。通过实验对比验证所提出的方法能够有效处理复杂Web页面结构，且聚类准确度高，时间复杂度低，是一种高质量的Web页面聚类方法。

**关键词**　Web测试；Web页面相似度；聚类
**中图分类号**　G354　　**DOI**：10. 3969/j. issn. 1672-9722. 2020. 05. 028

# A Marked Clustering Method for WEB Pages

JIAO Yongqiang[1]　WANG Weiyang[2]　SHANG Ying[2]
（1. AVIC China Aero-Polytechnology Establishment, Beijing　100028）
（2. Beijing University of Chemical Technology, Beijing　100029）

**Abstract**　Page clustering is an extremely effective method to reduce the number of redundant state in Web applications through clustering similar pages. Page clustering needs to analyze the similarity between two web pages, but traditional clustering methods can not accurately describe the complex page structure, high time complexity and low clustering accuracy. Therefore, this paper proposes an improved tree matching algorithm, which considers not only the structure of DOM tree, but also some attribute information of DOM, which makes this method better cope with complex web page structures. Experiments show that this paper proposes an effective page clustering method, which greatly reduces the clustering time and improve the accuracy.

**Key Words**　Web testing, Web page similarity, clustering
**Class Number**　G354

## 1　引言

随着互联网的飞速发展，Web应用以其易于开发与升级、扩展性好、系统灵活性强等优点，获得了越来越多企业的青睐。但Web应用程序相较于传统应用程序，拥有更独特的结构和功能，传统软件测试技术难以对Web应用进行有效的测试。

Web页面是构成Web应用的基本元素，是用户与Web应用交互的媒介。传统的Web应用为每个页面绑定了一个唯一的URL，因此，页面可以用URL表示。然而，在Web 2.0中，为丰富交互及响应，提升用户体验，JavaScript（JS）和DOM（Document Object Model）广泛使用。因此，Web页面的改变不再由URL决定，而是由DOM的动态改变决定。即Web页面表示为DOM树，JS代码执行过程中动态改变DOM树，从而使得页面发生变化。A. Nederlof等的研究表明，在Web 2.0应用中，平均每个URL对应16个DOM页面[1]。由此可以看出，DOM微小变化可使Web页面急剧扩充，导致页面集合十分复杂与庞大，增大了Web应用测试的难度。

在Web应用测试过程中，由于一个Web应用中大量页面具有类似的DOM树结构[2-4]，而对这些相似Web页面单独生成测试用例进行测试降低了

* **收稿日期**：2019年11月11日，**修回日期**：2019年12月5日
**基金项目**：国家自然科学基金项目（编号：61672085）资助。
**作者简介**：焦永强，男，硕士，研究方向：软件测试。王维扬，男，硕士研究生，研究方向：软件测试。尚颖，女，博士，副教授，研究方向：软件测试、信息聚合、算法优化。

测试生成效率。为减少相似Web页面对测试生成的影响，大多采用Web页面聚类方法，即将相似页面聚为同类，类内页面作为一个页面进行测试，以减少测试的时间开销。

目前，Web页面聚类方法主要有三类：面向URL的页面聚类[5]、基于页面超链接的页面聚类[6]和基于DOM结构的页面聚类方法[7~11]。由于JS和Ajax技术的使用，页面超链接和URL不能很好地表示Web页面，因此，其对应的页面聚类方法应用领域有限。当前的研究主要集中在基于DOM的页面聚类方法。文献[7]最早提出基于DOM树的页面聚类方法，该方法利用树编辑距离来度量两页面之间的相似度，并基于此对页面进行聚类，然而树编辑距离的计算复杂度高，且该方法未考虑由于页面内容不同导致的相似Web页面，因此，聚类准确度不高。文献[8]提出了一种基于DOM结构和样式相结合的页面聚类方法，采用树编辑距离作为结构相似度的度量准则，同时考虑页面样式，如布局/颜色/字体等的相似度对页面进行聚类。该方法聚类准确度有所提升，但其计算复杂度很高。

文献[9]提出了Bag of Xpath模型，把页面表示为包含当前页面元素位置索引的Xpath集合，通过计算两个页面Xpath集合元素之间的差异来度量二者之间的相似度，进而实现页面聚类，该方法虽然降低了计算复杂度，但其聚类准确度不高。文献[10]通过深度遍历将两个页面的DOM结构转换成一个元素节点标签序列，比较两个页面的节点标签序列来计算页面之间的相似度，同样地，该方法的页面聚类准确度不太理想。

综上，目前Web页面聚类准确率较低而时间复杂度较高。分析其原因，是由于1）上述方法大多仅考虑DOM树结构之间的相似性。具有相同DOM结构的页面可能表征的功能不同，在Web应用功能测试时不应被划分至同一类；2）现有的页面之间相似性度量方法不能准确区分不同页面，且其相似性计算及聚类算法的时间复杂度较高。

此外，Web页面属性为页面元素定义属性（如id、name和class），实现页面元素样式渲染以及事件绑定，为Web应用程序提供了多种定制化服务。可以看出，页面属性与Web页面功能密切相关。换言之，为区分表征不同功能的页面，页面元素的属性必不可少。

因此，本文提出一种新的页面聚类方法，不仅考虑了DOM结构之间的相似性，而且考虑了页面属性之间的相似性，同时给出了一种新的页面相似性度量方法。在此基础上，改进了现有的页面聚类算法，实现Web页面聚类。该方法不仅能够描述复杂的Web页面，且具有较低的时间复杂度和较高的页面聚类准确度。

## 2　Web页面DOM结构与节点属性

Web页面是构成Web应用的基本元素，DOM将页面表达为树结构，称为DOM树，DOM树上的节点类型包括元素节点、属性节点和文本节点，分别表示页面中的元素、文本和属性。为了表述方便，下文将“元素节点”统称为“节点”，“属性节点”统称为“属性”，“文本节点”统称为“文本”。所有DOM节点相互包含组成的树形结构构成了Web页面的基本结构，也被称为DOM结构。

### 2.1　DOM

DOM将Web页面表达为树结构，定义了访问和操作页面节点、文本及属性的标准方法。Web页面以标签（tag）来标识DOM节点，如图1所示。图1的树形结构表示了一个简单Web页面的DOM树，表示某教师管理系统的用户管理页面，该页面实现对教师的增删改等操作，其中浅灰线为属性，点线为文本，其余为节点。

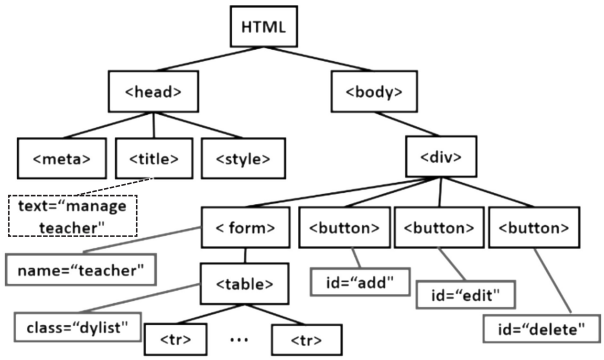



图1　DOM树

### 2.2　节点属性

Web页面的每一个节点都拥有若干属性，在构建网页时，节点属性必不可少，且每个属性都具有其作用。根据W3C标准，id属性用来唯一标识网页中的元素，如果两个Web页面中的两个节点具有相同的id，则它们有极大的概率是相同的节点。class属性用来标识一类元素，这一类元素往往具有相同的样式，因为CSS通常采用class（类）名来为节点定制样式。Id和class属性都有一个最重要的特性，即能够绑定事件，因此，二者与Web应用的行为密切相关。name属性比较特殊，表示一个节点的名字，常常用于form节点中。在Web应用中form节点被大量的使用，而在Web测试中不同的form

往往意味着与后台数据交互的不同,表示不同的功能,这对功能测试有很大的影响。

图2为某图书管理系统的书籍管理页面,实现对书籍借阅、查找、归还等操作。图1和2所示的两个DOM具有相同结构,但其div标签具有不同的id及name,且为不同的id绑定不同的事件。因此,即使这两个Web页面的DOM结构完全相同,也应被识别为不同的页面。

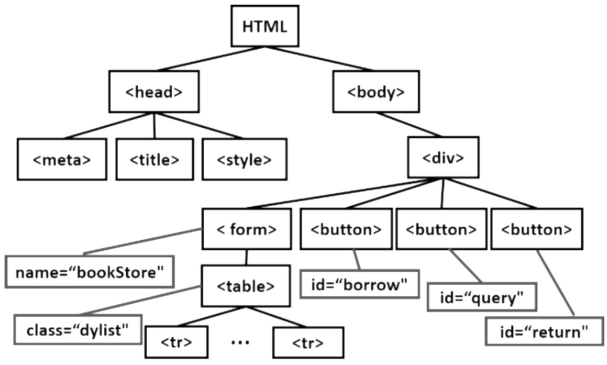



**图2 与图1具有相同DOM结构的DOM2**

显然,基于DOM结构的页面聚类方法对此类页面的聚类准确度较低。因此,为准确区分不同Web页面,在进行页面比较时,不仅需要考虑Web页面中DOM结构之间的相似性,属性之间的相似性比较在页面聚类中也十分重要。

## 3 Web页面相似性度量新方法

Web应用的页面中,具有相同DOM结构的页面可能表示不同的功能页,而在Web应用功能测试时不能被划分至同一类。因此,仅考虑DOM结构相似性的页面聚类方法会产生大量误报的情况。为提高页面聚类的准确性,本文同时考虑Web页面之间DOM结构和属性的差异,提出了新的页面相似度度量方法,并给出一种高效的改进树匹配算法来计算页面相似度。

本节首先定义页面之间的页面相似量定义;然后,给出改进树匹配算法计算页面相似量;最后,为比较多个Web页面之间的相似程度,基于页面相似量,给出页面之间相似度计算方法。

### 3.1 相似性度量指标

页面相似量用来度量两个Web页面,即两棵DOM树,之间的相似程度,由DOM树上各个节点之间的节点相似量求和得出。由上述分析可知,仅通过DOM结构之间的相似程度进行页面聚类是不充分的,页面属性也与Web应用功能密切相关,因此,本文同时考虑页面结构和属性之间的相似性,给出页面相似性度量方法,以准确识别相似页面。

为度量节点及属性之间的相似量,我们针对DOM树定义四种相似基量:Tag相似基量 $\lambda_{tag}$ 、ID相似基量 $\lambda_{id}$ 、Name相似基量 $\lambda_{name}$ 和Class相似基量 $\lambda_{class}$ ,相似基量表示节点相似量度量时节点及属性的相似量权重。

两棵DOM树根节点标签tag值相似基量为式(1)。

$$\lambda_{tag}=1 \tag{1}$$

节点各属性(id、name、class)的相似基量如式(2~4)。

$$\lambda_{id}=\frac{1}{\left(|ID_1|+|ID_2|\right)\Big/\left(|T_1|+|T_2|\right)} \tag{2}$$

$$\lambda_{name}=\frac{1}{\left(|NAME_1|+|NAME_2|\right)\Big/\left(|T_1|+|T_2|\right)} \tag{3}$$

$$\lambda_{class}=\frac{1}{\left(|CLASS_1|+|CLASS_2|\right)\Big/\left(|T_1|+|T_2|\right)} \tag{4}$$

其中 $|ID_1|$ 和 $|ID_2|$ 分别表示两个DOM树中具有id属性的节点个数,name和class同理,$|T_1|$ 和 $|T_2|$ 分别表示两个DOM树的节点总数。

算法一:两个Web页面DOM树 $T_1$ 和 $T_2$ 页面相似量计算

输入:两个Web页面DOM树 $T_1$ 和 $T_2$

输出:$T_1$ 相对于 $T_2$ 的相似量 $F(T_1,T_2)$

步骤1. 分别对 $T_1$ 和 $T_2$ 进行深度遍历得到它们的总结点个数 $|T_1|$ 和 $|T_2|$,分别具有id、name、class属性的节点个数 $|ID_1|$、$|NAME_1|$、$|CLASS_1|$、$|ID_2|$、$|NAME_2|$、$|CLASS_2|$;

步骤2. 通过式(2)~(4)计算各个属性的相似基量值 $\lambda_{id}$、$\lambda_{name}$、$\lambda_{class}$;

步骤3. 设 $T_1$ 的当前根节点为 $N_1$,$T_2$ 的当前根节点为 $N_2$,当前树的相似量为 $F(T_1,T_2)=0$ ;

步骤4. 通过式(5)计算的当前节点 $N_1$ 和 $N_2$ 的相似量 $f(N_1,N_2)$ , $F(T_1,T_2)=F(T_1,T_2)+f(N_1,N_2)$ ;

步骤5. 设 $T_1$ 和 $T_2$ 的子树集合分别为 $A=[T_{11},T_{12},T_{13},\cdots T_{1n}]$, $B=[T_{21},T_{22},T_{23},\cdots,T_{2m}]$,标识符 $i=1$,$j=1$,子树 $T_{1i}$ 相对于 $B$ 的相似量为 $F(T_{1i},B)=0$;

步骤6. 转步骤3进行递归计算得出 $T_{1i}$ 相对于 $T_{2j}$ 的相似量为 $F(T_{1i},T_{2j})$;

步骤7. $j=j+1$,判断如果 $j>m$,$F(T_1,T_2)=F(T_1,T_2)+F(T_{1i},B)$,$i=i+1$,$j=0$ ;否则 $F(T_{1i},B)=\mathrm{MAX}(F(T_{1i},B),F(T_{1i},T_{2j}))$,转步骤6;

步骤8. 判断如果 $i>n$,算法停止,输出 $F(T_1,T_2)$;否则转步骤6;

根据DOM特性,定义两个节点 $N_1$ 和 $N_2$ 之间的节点相似量:当tag值不同时,他们之间的节点相似量 $f(N_1,N_2)=0$,相同时相似量如式(5)。

$$f(N_1,N_2)=\lambda_{tag}+\lambda_{id}\times C_{id}+\lambda_{name}\times C_{name}+\lambda_{calss}\times C_{class} \tag{5}$$

其中当id值、name值相同时，$C_{id=}C_{name}$值为“1”，不相同时值为“0”；$C_{class}$计算公式如下：

$$C_{class}=\frac{2\times 相同class取值的个数}{两个节点class取值的总个数} \tag{6}$$

在节点相似量的基础上，给出页面相似量计算公式如下：假设两页面对应的两棵DOM树为$T_1$和$T_2$，其根节点分别为$N_1$和$N_2$，$T_1$和$T_2$子树的集合为$A=[T_{11},T_{12},T_{13},\cdots,T_{1n}]$，$B=[T_{21},T_{22},T_{23},\cdots,T_{2m}]$，则定义$T_1$相对于$T_2$的页面相似量为式(7)：

$$F(T_1,T_2)=f(N_1,N_2)+\sum_{i=1}^{n}\mathrm{MAX}_{j=1}^{m}\left(F(T_{1i},T_{2j})\right) \tag{7}$$

式(1~7)定义了节点之间的节点相似量及DOM树之间的页面相似量度量公式。为计算两棵DOM树之间的页面相似量，我们在现有树匹配算法[12]的基础上，增加了对页面属性信息的度量，提出了改进的树匹配算法，如算法一所示。

### 3.2　相似度定义

页面相似量是一个绝对值，它在一定程度上能反映两个页面的相似程度，但是在多个网页之间并没有可比性，因此，本文将页面相似量进行归一化，记为页面相似度，并用相似度来进行Web页面间的差异性比较。对于给定的两个Web页面，它们对应的DOM树分别为$T_1$和$T_2$，$|T_1|$和$|T_2|$分别表示两者的节点个数，假设二者包含id属性的节点总个数为$M$，包含name属性的节点总个数为$N$，包含class数属性的节点总个数为$K$，则它们之间的相似度定义为式(8)：

$$Similar(T_1,T_2)=(F(T_1,T_2)+F(T_2,T_1))/[(|T_1|+|T_2|)+\lambda_{id}*M+\lambda_{name}*N+\lambda_{class}*K] \tag{8}$$

其中$F(T_1,T_2)$，$F(T_2,T_1)$分别表示$T_1$相对于$T_2$的相似量及$T_2$相对于$T_1$的相似量，$\lambda_{id}$、$\lambda_{name}$和$\lambda_{class}$的计算如式(2~4)。

## 4　标记聚类算法

Web页面集合庞大，页面多种多样且结构复杂，因此，无法预先判断最终聚类的个数，这使得一些传统的聚类算法如k-means不再适用于Web页面聚类[13]。凝聚层次聚类(Hierarchical Agglomerative Clustering，HAC)方法是一种常用的层次聚类算法，该方法无需预先设定类簇个数，常被应用于Web页面聚类中[14~15]。基础凝聚层次聚类HAC算法具有$o(n^3)$的时间复杂度，改进的凝聚层次聚类算法[16]，能达到的最好的时间复杂度是$o(n^2)$。本文在改进的HAC基础上，提出了标记聚类算法(Marked Clustering，MC)，即在聚类的同时进行页面标记，在理想的情况下最低能达到$o(n)$的时间复杂度。

### 4.1　MC聚类算法基本思想

为了在保证聚类准确性的同时减少聚类所用的时间，本文基于Web页面的特性提出标记聚类算法。该算法的核心思想是在计算Web页面间相似度之后，对页面进行聚类标记，即当Web页面之间的相似度超过设定阈值，则将两个Web页面标记为同一类；针对已经标记的Web页面不再进行后续的Web页面比较；且各类仅取任一页面作为当前类中所有页面的代表，当判断其他Web页面是否归属于此类时，仅与类中的代表页面进行相似度比较即可。

### 4.2　算法描述

标记聚类MC主要有以下五个步骤，算法流程如图3所示。

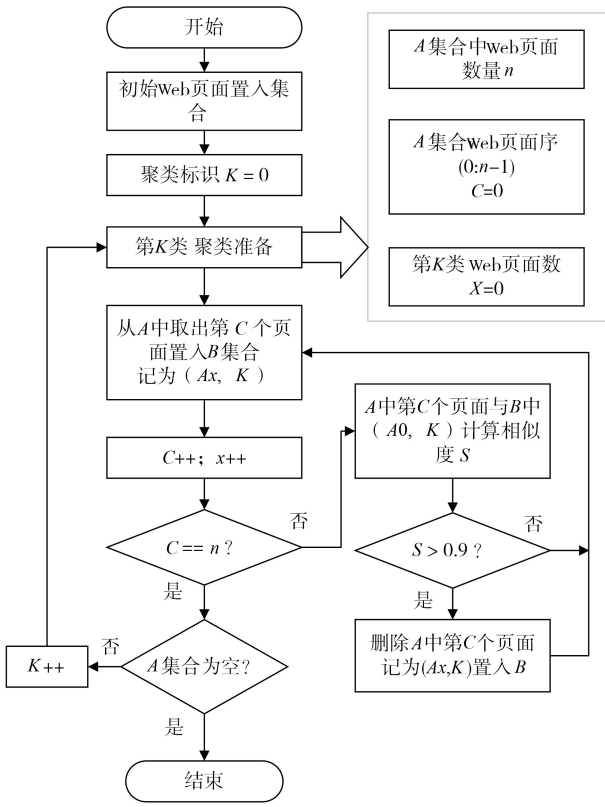



图3　MC算法流程

1)初始化两个集合$A$、$B$，其中$A$是所有Web页面的集合，$B=\Phi$；聚类标记$k=0$；

2)从$A$中选择一个元素$a$，将$[a,k]$添加到$B$中，然后从$A$中删除$a$；

3)从$A$中顺序选择一个元素$b$，通过改进树匹配算法计算$a$与$b$之间的页面相似度，如果相似度大于阈值则将$[b,k]$添加到$B$中，然后从$A$中删除

$b$;否则进行下一步;

4)判断是否遍历$A$中所有元素,如果是进行下一步,否则转3);

5)判断$A$是否为空,如果不为空则令$k=k+1$,转2)。

最终得到$B$集合即为聚类结果,$B$中每个元素均为二元组,表示Web页面及其所属的类簇的标号。

标记聚类MC算法的时间复杂度是$o(n^2)$,但是由于每一次循环都会减少下一次需要比较的页面数量,事实上程序运行时间会大大降低。且最终聚类的数量越少,该方法的时间复杂度越小。在理想的情况下最少能达到$o(n)$的时间复杂度。

## 5 实验

为了验证本文方法的有效性,我们提出了两个研究问题如下:

1) Web页面相似性度量方法的有效性如何?改进树匹配算法是否优于其他树匹配算法?

2) 标记聚类算法MC是否能在不影响聚类结果的前提下提高聚类效率?

### 5.1 实验对象与环境配置

实验采用两个开源的Web应用e107和wordpress作为被测程序。实验在Intel 酷睿i5(3470 3.4GHZ) 4核CPU、内存8GB、Win10操作系统、Python 3.6.4和htmlParser环境下进行。本文选用常用的简单树匹配算法作为相似性对比算法,HAC作为聚类对比算法,设计了以下两个实验。

实验一:分别使用简单树匹配和本文提出的改进树匹配算法完成页面相似性度量,聚类方法均采用同一凝聚层次聚类方法进行页面聚类。

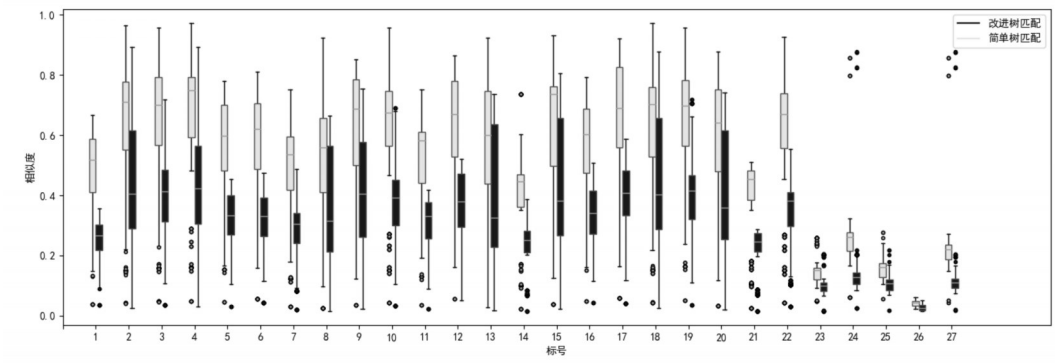
实验二:相似性度量采用本文提出的改进树匹配的算法,聚类方法分别使用凝聚层次聚类和本文提出的标记聚类进行比较。

### 5.2 实验结果

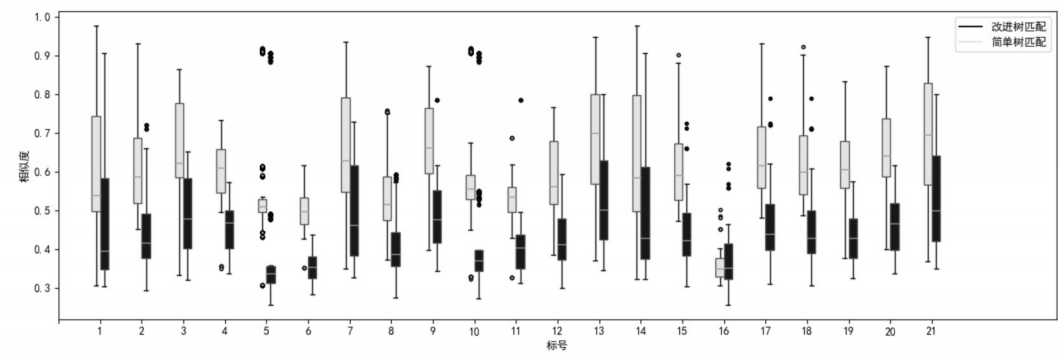
#### 5.2.1 相似度算法结果

为了比较两种相似度算法(改进树匹配和简单树匹配)的优劣,本实验收集了两个开源Web应用的100张Web页面,并对其进行人工聚类,然后将每一类的页面与同一类的页面以及其他类的页面进行相似度计算,比较他们的相似度值。对于同类页面,两种算法均得到了大于0.9的相似度值,这说明对于相似页面,两种算法都能很好的比较其相似度。但对于不同类的页面,实验结果如图4所示。

从图4可以看出,本文提出的改进树匹配算法对于属于不同类的页面计算得到的相似度明显小于简单树匹配算法的结果,而且均小于0.9,这说明本文的改进树匹配算法区分不同类页面的能力强于简单树匹配算法。




(a) e107




(b) wordpress

图4 两种算法的页面相似度计算比较结果

表 1　e107

| | 人工标记类数 | 聚类数量 | 正确标记类数 |
|---|---|---|---|
| NEW | 28 | 28 | 28 |
| 匹配树 | 28 | 21 | 18 |

表 2　wordpress

| | 人工标记类数 | 聚类数量 | 正确标记类数 |
|---|---|---|---|
| NEW | 21 | 21 | 21 |
| 匹配树 | 21 | 17 | 13 |

为了直观展示两种相似度算法对聚类结果的影响，本部分还给出了分别使用两种相似度度量算法结合同一层次聚类算法得到的聚类结果，如表 1，2 所示，本文提出的改进树匹配算法准确率和召回率都明显优于简单树匹配算法。这是由于简单树匹配将大量的不属于同一类但是却有相似 DOM 结构的网页聚为一类，使得该算法的召回率极低。经过进一步的分析发现，由于 Web 应用中大量使用同一框架和 form 表单，这使得简单树匹配算法聚类失误，但本文提出的改进树匹配算法考虑了更多的属性信息，从而得到了更好的聚类效果。

5.2.2　聚类算法结果

为比较两种不同的聚类算法的效率，采用相同的页面相似性度量方法，即改进树匹配算法，不同的页面聚类算法对上述两个 Web 应用进行了页面聚类实验。其中凝聚层次聚类 HAC 算法中类之间的相似度采用如式(9)进行计算：

$$Sim(cluster1, cluster2) = \min_{\substack{0<i<|cluster1| \\ 0<j<|cluster2|}} (similar(T_i, T_j)) \tag{9}$$

两种聚类算法对 Web 页面聚类的实验结果如表 3 和表 4 所示。

表 3　e107 聚类算法比较

| | 实际类数 | 聚类数量 | 正确标记的类数 | 准确率 | 召回率 | 运行时间(ms) |
|---|---|---|---|---|---|---|
| HAC | 28 | 28 | 28 | 100% | 100% | 180437 |
| MC | 28 | 28 | 28 | 100% | 100% | 52957 |

表 4　wordpress 聚类算法比较

| | 实际类数 | 聚类数量 | 正确标记的类数 | 准确率 | 召回率 | 运行时间(ms) |
|---|---|---|---|---|---|---|
| HAC | 21 | 21 | 21 | 100% | 100% | 245496 |
| MC | 21 | 21 | 21 | 100% | 100% | 43233 |

观察上表可以明显看出本文提出的标记聚类算法 MC 并没有影响聚类的结果，但是却明显减少了聚类时间，提高了页面聚类效率。具体来说，对于 e107 效率提升了 3.4 倍，而对于 wordpress 效率提升了 5.6 倍。因此，本文提出的标记聚类算法效率更高。

## 6　结语

在 Web 测试中，为解决现有方法在页面聚类时准确率低及效率不高的问题，本文提出了一种改进树匹配算法，不仅考虑 Web 页面结构信息还考虑部分属性信息，通过该算法来计算 Web 页面之间的相似度显著提高了聚类的准确性。同时为了解决传统聚类算法耗时长的问题，提出一种更为简单有效的标记聚类算法 MC。实验证明，本文的聚类算法在不影响聚类的准确性的前提下显著地降低了聚类所用的时间。

### 参 考 文 献


[1] A. Nederlof, A. Mesbah, and A. van Deursen. Software engineering for the web: The state of the practice[C]//In Proceedings of the ACM/IEEE International Conference on Software Engineering, Software Engineering In Practice (ICSE SEIP), ACM, 2014:4-13.

[2] S. Sabharwal and P. Bansal. A model based approach to test case generation for testing the navigation behavior of dynamic web applications[C]//Contemporary Computing (IC3), Sixth International Conference,2013:213-218.

[3] S. Sabharwal, P. Bansal and M. Aggarwal. Modeling the Navigation Behavior of Dynamic Web Applications[J]. International Journal of Computer Applications (0975-8887) 2013, 20-27.

[4] Polpong J, Kansomkeat S. Syntax-based test case generation for Web application [C]//Computer, Communications, and Control Technology (I4CT), 2015 International Conference on. IEEE, 2015:389-393.

[5] Mehta B, Narvekar M. DOM tree based approach for Web content extraction[C]// International Conference on Communication, Information & Computing Technology. IEEE, 2015: 1-6.

[6] Alarte J, Insa D, Silva J, et al. Site-Level Web Template Extraction Based on DOM Analysis[C]// International Andrei Ershov Memorial Conference on Perspectives of System Informatics. Springer International Publishing, 2015: 36-49.

[7] Zhang K, Shasha D. Simple fast algorithms for the editing distance between trees and related problems[J]. SIAM journal on computing, 1989, 18(6): 1245-1262.

[8] Gowda T, Mattmann C A. Clustering Web Pages Based on Structure and Style Similarity (Application Paper)[C]// IEEE, International Conference on Information Reuse and Integration. IEEE, 2016: 175-180.


(下转第 1200 页)

[3] Schnabel R, Wahl R, Klein R. Efficient RANSAC for Point-Cloud Shape Detection[J]. Computer Graphics Forum, 2010, 26(2):214-226.

[4] Larson J,Trivedi M. Lidar Based Off-road Negative Obstacle Detection and Analysis[J]. Conference Record IEEE Conference on Intelligent Transportation Systems, 2011.

[5] Lalonde J, Vandapel N, Huber D F, et al. Natural terrain classification using three-dimensional ladar data for ground robot mobility [J]. Journal of Field Robotics, 2010, 23(10):839-861.

[6] Guo C, Sato W, Han L, et al. Graph-based 2D road representation of 3D point clouds for intelligent vehicles[C]// Intelligent Vehicles Symposium. IEEE Xplore, 2011: 715-721.

[7] Chen X, Ma H, Wan J, et al. Multi-view 3D Object Detection Network for Autonomous Driving[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:6526-6534.

[8] Li B, Zhang T, Xia T. Vehicle detection from 3d lidar using fully convolutional network[J]. arXiv preprint arXiv: 1608.07916, 2016.

[9] Velas M, Spanel M, Hradis M, et al. CNN for very fast ground segmentation in velodyne LiDAR data[C]//2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2018: 97-103.

[10] Wu Z, Song S, Khosla A, et al. [IEEE 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Boston, MA, USA (2015.6.7-2015.6.12)] 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - 3D ShapeNets: A deep representation for volumetric shapes,2015:1912-1920.

[11] Maturana D, Scherer S. 3D Convolutional Neural Networks for landing zone detection from LiDAR[C]// IEEE International Conference on Robotics and Automation. IEEE, 2015:3471-3478.

[12] Maturana D, Scherer S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2015:922-928.

[13] Sedaghat N, Zolfaghari M, Amiri E, et al. Orientation-boosted voxel nets for 3d object recognition[J]. arXiv preprint arXiv:1604.03351, 2016.

[14] Charles R Q, Su H, Mo K, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation [C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:77-85.

[15] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[C]// Advances in Neural Information Processing Systems, 2017: 5099-5108.

[16] Li Y, Bu R, Sun M, et al. PointCNN: Convolution On X-Transformed Points[C]//Advances in Neural Information Processing Systems,2018: 828-838.

(上接第1153页)

[9] Joshi S, Agrawal N, Krishnapuram R, et al. A bag of paths model for measuring structural similarity in Web documents[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003: 577-582.

[10] 尤枫, 张雅峰, 赵瑞莲,等. 基于页面聚类的Web应用测试方法研究[J]. 计算机工程与应用, 2018,54(5): 51-56.
YOU Feng, ZHANG Yafeng, ZHAO Ruilian, et al. Page cluster based research on Web application test method [J].Computer Engineering and Applications, 2018, 54(5) :51-56.

[11] Lanotte P F, Fumarola F, Malerba D, et al. Exploiting Web Sites Structural and Content Features for Web Pages Clustering[C]//International Symposium on Methodologies for Intelligent Systems. Springer, Cham, 2017: 446-456.

[12] 何昕, 谢志鹏. 基于简单树匹配算法的Web页面结构相似性度量[J]. 计算机研究与发展, 2007, 44(z3): 1-6.
HE Xin, XIE Zhipeng. Structural Similarity Measurement of Web Pages Based on Simple Tree Matching Algorithm [J]. Journal of Computer Research and Development,2007,44(z3):1-6.

[13] Santini, Marina. Advantages & disadvantages of k-means and hierarchical clustering (unsupervised learning) [EB/OL].URL: http://santini. se/teaching/ml/2016/Lect_10/10c_Unsupervise dMethods. pdf(Accesed 17.04. 2019) (2016).

[14] Harper, Simon, et al. DOM block clustering for enhanced sampling and evaluation[C]//Proceedings of the 12th Web for All Conference. ACM, 2015.

[15] Ibrahim, R. , S. Zeebaree, and K. Jacksi. Survey on Semantic Similarity Based on Document Clustering [J]. Adv. Sci. Technol. Eng. Syst. J 4.5 (2019): 115-122.

[16] F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms[J]. Computer Jonrnal, 1983, 26(4):354-359.