一种改进的树路径模型在网页聚类中的研究

王亚普[1] 王志坚[1,2] 叶 枫[1,2]
(河海大学计算机与信息学院 南京 211100)[1] (南京航空航天大学计算机科学与技术学院 南京 210016)[2]

摘 要 相似度计算是文本挖掘的基础,也是信息提取过程的关键步骤。对于结构复杂的网页,当前基于传统树路径模型的相似度计算方法在准确性上尚不完善。传统树路径模型未考虑路径出现的先后顺序,并且比较路径相似度时用的是完全匹配,难以在不完全匹配时更精确地描述路径之间的相似度。因此,从网页结构相似度入手,提出了一种改进的树路径模型。该模型充分考虑了兄弟节点之间的关系、路径位置以及路径权重,弥补了传统树路径模型无法表达文档结构和层次信息的缺陷。实验结果表明,该模型提高了识别网页结构相似性的能力,既能对结构差别较大的网页进行良好的区分,又能较好地反映来自同一模板的网页之间的差异性,同时在网页聚类中具有更优的效果。

关键词 信息提取,网页结构,相似度,树路径模型,聚类
中图法分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.5.022

Research of Improved Tree Path Model in Web Page Clustering

WANG Ya-pu[1] WANG Zhi-jian[1,2] YE Feng[1,2]
(College of Computer and Information, Hohai University, Nanjing 211100, China)[1]
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[2]

Abstract Computing the similarity is the basis of text mining, and also the crucial step of information extraction. When tackling the Web pages with complex structure, the accuracy of computing the similarity based on traditional tree path model is not perfect. Traditional tree path model will not take the sequence of the paths in consideration and compare the similarity of paths by using perfect matching. It cannot describe the similarity between paths accurately when it is not a perfect matching. Therefore, the paper introduced the structural similarity Web at first, and then proposed a tree path model. This model takes fully account of the relationship between the siblings, the path location and the path weights, and makes up for the defect of the traditional tree path model which cannot express both document structure and hierarchical information. The experiment result shows that the model improves the recognition ability of Web pages structural similarity. It not only can better distinguish the Web pages which have large structure difference, but also effectively reflects the difference between the Web pages with the same template, also has a better effect in the Web page clustering.

Keywords Information extraction, Web page structure, Similarity, Tree path model, Clustering

1 引言

统计结果表明,Internet每天网页的增加量约为一百万,每两年网页数量将增加3倍,其增长速度远超摩尔定律增长速度。Web数据通常以诸如HTML、XML等半结构化的方式存储,具有灵活性和异构性。如何从网页中高效、准确地提取用户感兴趣的信息,成为当前信息检索面临的挑战之一。网页结构相似性度量是Web信息提取预处理阶段的重要过程,它广泛应用于XML文档内容提取[1,2]、模式抽取、基于聚类的搜索引擎、查询优化等方面。若能够快速、准确地识别出结构相似的网页,并以此为依据对网页进行聚类,就可以有效地简化数据内容并发现其中的分类模式。因此,研究如何识别相似度较高的网页具有重要的研究价值和意义。当前的研究现状表明,从不同角度对网页进行聚类的准确度和可视化展示远未达到用户多样化的需求。本文尝试以传统树路径模型作为理论基础,对其在相似度计算方面的缺陷进行改进,利用改进的树路径模型设计了一个网页聚类的方案,并对聚类效果进行评价,进一步验证了方案的可行性。

HTML是一种通用的、用于描述网页文档的半结构化语言。然而,在半结构化信息中存在一定的结构化信息,获取这些结构化信息常常依赖于HTML语言的标记。一个HTML文档可以用一棵有序的带标记的文档对象模型(Document Object Model,DOM)树来表示,树中的每个节点对应文档中的一个元素,树的每条边对应两个元素之间的关系,因此网页

到稿日期:2014-02-18 返修日期:2014-04-21 本文受江苏水利科技项目:"智慧河流"研究及其在六合滁河管理中的应用(2013025),河海大学中央高校基本科研业务费项目(2009B21614)资助。
王亚普(1990-),男,硕士生,主要研究方向为大数据挖掘,E-mail:wangyapu0714@gmail.com;王志坚(1958-),男,教授,博士生导师,主要研究方向为软件理论,E-mail:zhjwang@hhu.edu.cn(通信作者);叶 枫(1980-),男,讲师,主要研究方向为云计算。

之间的相似性可以用对应 DOM 树之间的相似性进行度量。在国内外研究[3-8]中，有很多用于度量树之间的相似性的方法，例如树路径和树的编辑距离已经被广泛使用，但是这些方法在某方面均存在一定的不足，有待进一步的改进和完善。

本文第 2 节介绍了计算网页结构相似性已有的研究工作；第 3 节重点阐述了改进的树路径模型以及网页结构相似性的计算方法；第 4 节介绍了基于网页相似度的层次聚类算法以及聚类的评估指标；第 5 节通过实验验证了模型的有效性和可行性；最后对本文进行总结，并展望未来工作。

2 相关工作

在国内外的研究中，用于度量网页之间的相似性的方法可大体分为 3 类：1)基于超链接的分析，PageRank 算法是其中最成功的代表算法之一；2)基于文本信息的分析；3)基于网页结构的分析。不同类型的方法有着各自的适用范围，本文针对网页结构的相似度分析整理了以下相关工作。文献[3]提出了简单树匹配算法，其不但保留了基本的信息，而且记录了重复模式的相关信息，以便计算两棵树的最大匹配。但该算法不允许节点替换和跨层的匹配，对节点的顺序要求严格，且算法复杂度较高。Tai Kc[4]最早阐述了两棵树的基于编辑距离的结构相似度计算方法，其基本思想是使用编辑操作将一棵树转化为另一棵树的代价定义为两棵树的距离。该方法应用非常广泛，在处理节点数较少的 DOM 树时效果比较好，其缺陷是具有较高的时间复杂度，且网页所对应的 DOM 树往往节点较多，遍历 DOM 树中的每个节点是不切实际的。文献[5]提出了基于节点统计特征的网页结构相似性度量方法，根据网页标记的频率分布特征来计算两个网页之间的相似度，该方法简单高效，但准确率较低。S. Joshi[6]等人提出了树路径模型，该模型利用 DOM 树中对应的路径来表示文档的结构信息，并给出了相应的相似度计算方法。树路径模型比树的编辑距离要简单，而且时间复杂度低很多。虽然树路径模型弥补了树模型的部分缺陷，但该方法也有不足之处，即只考虑到父子关系，忽略了兄弟节点之间的关系、路径位置以及各路径的权重，而且不能很好地处理路径重复匹配的问题，从而造成准确度的降低。

对 HTML 文档的建模很大程度上决定了相似性的度量方法，因此 HTML 文档的模型描述和相似性的计算公式是两个研究重点。首先，需要对相关的结构信息建立模型，并根据此模型给出相似性的度量公式。本文针对传统树路径模型的相似度算法所存在的问题，结合考虑父子节点、路径位置以及各路径权重的情况下，提出了一种改进的树路径模型。该模型从不同角度衡量了网页结构的相似度，提高了网页结构相似度的识别能力。

3 网页结构相似度的计算

3.1 网页预处理

在对 HTML 文档解析的过程中，常常会遇到下述的问题：1)元素标签丢失。标签的作用是定位元素的位置，HTML 的元素标签主要分为必须有结束标签、无需结束标签和结束标签可有可无 3 类。第一类元素可以按其出现的顺序的倒序将结束插入〈/html〉标签之前；第二类元素在起始标签之后直接插入结束标签；针对第三类元素可以建立相应的广义结束标记表[7]，可以将结束标记插入其后出现的第一个广义结束标记之前。例如：〈thead〉标签可以在之后出现的第一个〈thead〉、〈tbody〉、〈tfoot〉、〈/table〉之前插入〈/thead〉。考虑到文档遍历的复杂性和第三类元素出现的情况极少，本文只针对前两种情况进行处理。2)元素节点的交叉。解决元素节点交叉情况的基本思路是搜索包含交叉元素节点结束标记的元素，从该元素的结束标记处分开，在其结束标记的前后各加上开始标签和结束标签。例如〈a〉〈b〉〈/a〉〈/b〉，消除元素交叉后的处理结果为：〈a〉〈b〉〈/b〉〈/a〉〈b〉〈/b〉。对应的数据结构和算法伪代码如表 1 所列。3)注释与脚本标签的嵌套。一般来说，脚本和注释对于网页结构分析来说并无太大关系，所以在解析过程中暂不考虑这部分内容。

表 1 消除交叉元素的伪代码

```
public class DOMNode {
        private String nodecontent;//节点内容
        private int nodetype;//节点类型：开始标记、结束标记、注释标记、文本
        DOMNode nextnode;//下一个节点
}
public void crossRemove(){
DOMNode node;
node =getRootElement();//获取根节点
//初始化堆栈;
while (node != null) {
switch(node. getNodetype()) {
case 注释：
case 文本：
      node = node. getNextnode();
      break;
case 开始标记：
      Push(node. getNodecontent());//节点内容压栈
      node = node. getNextnode();
      break;
case 结束标记：
      if (栈顶元素是否与该结束标记对应){
        Pop();//出栈   node = node. getNextnode();
      }
   else{
在该标记前后插入栈顶元素对应的开始标记和结束标记;
            Pop();//出栈
          }
          break;
        }
      }
}
```

3.2 网页相似度的计算

3.2.1 HTML 文档的 DOM 树表示方法

DOM 定义了 HTML 文档和 XML 文档的逻辑结构。HTML 文档中的所有节点构成了一棵 DOM 树[8]，每棵 DOM 树都有一个 HTML 元素作为根节点。图 1 给出了一个简单的 HTML 文档和对应的 DOM 树。

构建 DOM 树的方法：以〈html〉作为根节点，对 HTML 文档向后进行搜索，若获取到的标签是起始标记，则继续向后搜索其对应的结束标记。从根节点构造出一个新的子节点 newnode，以 newnode 作为当前子树的根节点，对该起始标记和结束标记所包含的内容继续进行分析。整个 DOM 树的构造过程是一个递归过程，其递归函数的代码如表 2 所列。

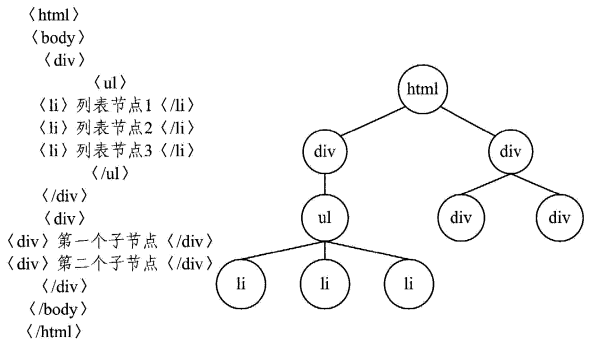

图 1　HTML 文档与对应的 DOM 树

表 2　DOM 树构建的递归函数

```
public void getElementList(Element element){
    List elements = element.elements();
    if (elements.size() == 0) {
      //没有子元素
      String xpath = element.getPath();
      String value = element.getTextTrim();
      elemList.add(new Leaf(xpath, value));
    }
    else {
      //有子元素
    for (Iterator it = elements.iterator(); it.hasNext();) {
        Element elem = (Element) it.next();
        //递归遍历
        getElementList(elem);
      }
    }
}
```

3.2.2　改进的树路径模型

DOM 树的构建过程实际也是树路径模型的建立过程，树路径是指从根节点到达叶子节点所经过的所有节点的序列，每棵 DOM 树均可用从根节点到达所有叶子节点的路径集合表示，路径的个数等于 DOM 树中的叶子节点数。例如图 2 中两棵 DOM 树所对应的树路径集合分别为：{abc，abc，ad}、{abc，aed}。

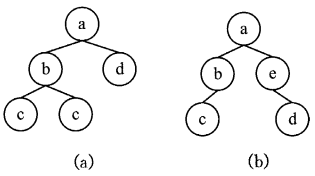

图 2　一组 DOM 树实例

相比于树编辑距离算法，S. Joshi 等人提出的树路径模型以及相似度的计算方法的复杂度要低得多。此模型中每个路径只需搜索对方路径集合中以根节点为起始节点且匹配元素最多的路径，两条路径的最长公共序列长度与较长的路径长度之比作为这两条路径的相似比。对于两棵 DOM 树 t_1 和 t_2，树路径集合分别表示为：

$paths(t_1)=\{p_{11},p_{12},\cdots,p_{1n}\}$

$paths(t_2)=\{p_{21},p_{22},\cdots,p_{2m}\}$

t_1 和 t_2 的相似度计算公式为：

$$sim(t_1,t_2)=\frac{1}{2}\times\left[\frac{1}{pnum(t_1)}\sum_{i=1}^{pnum(t_1)}\max(sim(p_{1i},p_{2j}))+\frac{1}{pnum(t_2)}\sum_{i=1}^{pnum(t_2)}\max(sim(p_{2i},p_{1j}))\right] \quad (1)$$

其中，$pnum(t_1)$、$pnum(t_2)$分别代表两棵 DOM 树中包含树路径的个数，$\max(sim(p_{1i},p_{2j}))$代表路径 p_{1i}相对于 DOM 树 t_2 的最佳匹配路径与 p_{1i}之间的相似度。网页结构相似度的值越接近 1，表示两个网页结构越相似，图 2 中两棵 DOM 树按式(1)计算其相似度为：

$$sim(t_1,t_2)=\frac{1}{2}\times\left[\frac{1}{3}\times(1+1+\frac{1}{3})+\frac{1}{2}\times(1+\frac{1}{3})\right]=\frac{13}{18}$$

然而上述方法只考虑到父子关系，忽略了兄弟节点之间的关系、路径位置以及各路径的权重。由于 HTML 文档中可能存在许多重复路径，并且一条路径可能有多条最佳匹配路径，因此该模型不能避免重复计算和判断多条最佳匹配路径中哪条路径更加符合实际情况。例如图 3 中 DOM 树(a)的每条路径在 DOM 树(b)中的最佳匹配路径均为同一条路径，这时就需要弱化它们之间的相似度。为了解决上述问题，本文提出了一种改进的树路径模型，其弥补了传统树路径模型无法很好地表达 HTML 文档结构信息和层次信息的缺陷。

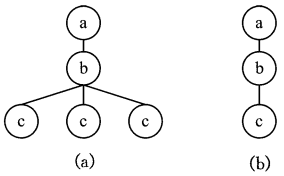

图 3　一组 DOM 树实例

DOM 树 t_1 和 t_2 中的两条树路径可定义为如下四元组：

$p_1=\{(v_{11},v_{12},\cdots,v_{1n}),(b_{11},b_{12},\cdots,b_{1n}),f_1,(lc_{11},lc_{12},\cdots,lc_{1f_1})\}$

$p_2=\{(v_{21},v_{22},\cdots,v_{2m}),(b_{21},b_{22},\cdots,b_{2m}),f_2,(lc_{21},lc_{22},\cdots,lc_{2f_2})\}$

其中，$(v_{11},v_{12},\cdots,v_{1n})$代表该路径所经历的标签序列，$(b_{11},b_{12},\cdots,b_{1n})$中 b_{1i}代表节点 v_{1i}在其兄弟节点中出现的次数，f_1 表示此路径在 HTML 文档中出现的次数，$(lc_{11},lc_{12},\cdots,lc_{1f_1})$表示该路径在 HTML 文档中出现的位置。在图 2 中 DOM 树(a)所包含的路径 *abc* 可以表示为如下四元组：

$p=\{(a,b,c),(1,2,1),2,(1,2)\}$

路径 p_1 和 p_2 的相似度由 3 个因素决定：

$sim(p_1,p_2)=sim_{content}\times sim_{length}\times sim_{location}$

第一个因素是路径 p_1 和 p_2 在内容和结构上的相似性。表 3 列出了几种情况下的树路径匹配情况。可以看出传统的树路径模型使用的是完全匹配，但是这种方式会造成大量相似节点无法识别，导致识别的准确度较低。改进后的树路径匹配模型采用最长相似子序列作为最佳匹配串，尽可能地保留了网页的结构信息。

表 3　改进后的路径匹配方法与传统方法的对比

序号	p_1	p_2	传统模型	改进模型
1	abc	abd	ab	ab
2	abcd	abec	ab	abc
3	abc	dbc	无	无

假设路径 p_1 和 p_2 在利用改进后的匹配方法得到的相似子序列为 $z=(z_1,z_2,\cdots,z_k)$，在路径 p_1 和 p_2 中与序列 z 相符合的所有相似子串里，取相似子串中每个节点与根节点距离之和最小的子串作为它们的最佳匹配串，下标位置分别记为 $index_1=(l_{11},l_{12},\cdots,l_{1k})$、$index_2=(l_{21},l_{22},\cdots,l_{2k})$，最佳匹配串中每个节点在所有相似子串里重复出现的次数分别记为 $times_1=(g_{11},g_{12},\cdots,g_{1k})$、$times_2=(g_{21},g_{22},\cdots,g_{2k})$。路径 p_1 和 p_2 在内容和结构上的相似度可用以下公式进行衡量：

$$sim(p_1,p_2)=\frac{1}{k}\sum_{i=1}^{k}sim(v_{1l_i},v_{2l_i}) \quad (2)$$

其中，$sim(v_{1l_i},v_{2l_i})$受节点位置、节点频率的影响。从实际情

况考虑，HTML 文档中越靠近顶层的节点，其重要性往往越高，因此节点之间位置差距越大，它们的相似度也就越低。对于相同位置的节点，节点在所有相似子串里重复出现的次数以及在兄弟节点中出现的次数越高，说明该节点对文档结构的重要程度越大。式(3)为 $sim(v_{1l_i},v_{2l_i})$ 的计算公式：

$$sim(v_{1l_i},v_{2l_i})=\frac{1}{|depth(v_{1l_i})-depth(v_{2l_i})|+1}\times\frac{1}{2}\times\left(\frac{\min(g_{1l_i},g_{2l_i})}{\max(g_{1l_i},g_{2l_i})}+\frac{\min(b_{1l_i},b_{2l_i})}{\max(b_{1l_i},b_{2l_i})}\right) \tag{3}$$

第二个因素是路径 p_1 和 p_2 在长度上的相似性。两条路径的长度差异越大，它们之间的相似性就越小。sim_{length} 可以定义为：

$$sim_{length}=\frac{\min(length(p_1),length(p_2))}{\max(length(p_1),length(p_2))} \tag{4}$$

第三个因素是路径 p_1 和 p_2 在位置上的相似性。传统的树路径匹配模型未考虑一条路径在 HTML 文档中可能出现多次，这就造成了重复匹配的问题。本文利用路径在文档中的位置差异，弱化了这种情形下路径之间的相似度。式(5)给出了 $sim_{location}$ 的定义：

$$sim_{location}=1-\frac{1}{f_1+f_2}\times\frac{\sum_{k=1}^{f_1}dis(lc_{1k})+\sum_{k=1}^{f_2}dis(lc_{2k})}{\max(n,m)+1} \tag{5}$$

其中，$dis(l_{1k})$ 表示路径 p_1 在位置 l_{1k} 处与路径 p_2 的最近距离：

$$dis(lc_{1k})=\min(|lc_{1k}-lc_{21}|,|lc_{1k}-lc_{22}|,\cdots,|lc_{1k}-lc_{2k}|) \tag{6}$$

对于图 2 中两棵 DOM 树的路径 abc，根据式(6)，采用改进后的树路径模型计算其相似度为：

$$sim(p_1,p_2)=\frac{1}{3}\times(1+1+\frac{3}{4})\times\frac{3}{3}\times[1-\frac{1}{2+1}\times(\frac{0+1}{3}+\frac{0}{3})]=\frac{22}{27}$$

两棵 DOM 树之间的相似度最终可以根据式(1)获得，此时 $pnum(t_1)$、$pnum(t_2)$ 不再表示两棵 DOM 树中包含树路径的个数，而是代表树路径集合中元素的个数。

本文从以上 3 个不同的视角度量了树路径之间的相似性，将三者综合考虑使得文档相似度的计算更为准确。根据实际情况，可以为 3 个因素分别设置不同的权重。

4 基于网页结构相似度的层次聚类

聚类[9,10]是将数据划分为若干个子集的过程，得到的各个子集内数据对象具有较高的相似度，而各个子集间的数据对象则是不相似的或相似度较低的。由于在对网页集合进行聚类之前，无法预知聚类结果中簇的数目，因此本文采取层次聚类算法对网页集合进行聚类。

层次聚类[10]算法被广泛研究与应用，按照聚类方向的不同可以分为凝聚层次聚类和分裂层次聚类。层次聚类需要计算类与类之间的距离，有 5 个常用的簇间距离度量方法：最小距离法、最大距离法、中心法、类平均距离法、离差平方和法。本文采用凝聚层次聚类算法，对于簇间距离使用类平均距离进行度量。凝聚层次聚类采用自底向上的策略，首先将每个对象看作一个簇，然后将距离最近的两个簇合并成一个簇，重复此步骤，直到所有对象都在一个簇中，或者任意两个簇之间的距离均大于给定的阈值。平均距离法需要建立距离矩阵，定义簇间距离为类间元素两两距离的平均值，该方法倾向于合并差异较小的两个类，产生的分类具有一定的稳定性。针对网页集合进行聚类分析的具体步骤的流程如图 4 所示。

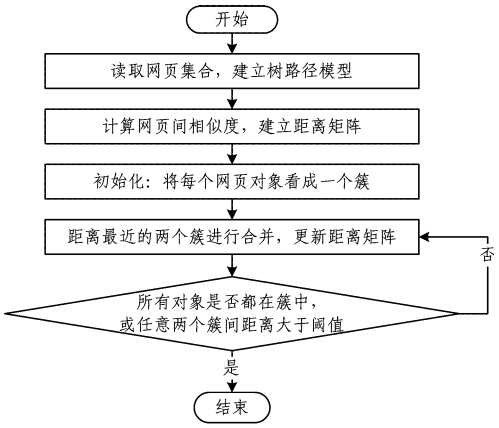

图 4 网页集合层次聚类流程

通常使用 F-measure[11]值作为评价聚类效果好坏的指标，它是信息检索领域一个常用的评价标准，其计算公式为：

$$F_\beta=\frac{(\beta^2+1)PR}{\beta^2P+R} \tag{7}$$

其中，β 是参数，P 是准确率(Precision)，R 是召回率(Recall)。当参数 $\beta=1$ 时，就是常见的 F1-measure。设网页集合聚类产生的簇集合为 $C=\{C_1,C_2,\cdots,C_n\}$，预定义的网页集合分类为 $D=\{D_1,D_2,\cdots,D_m\}$，聚类簇 C_i 对预定义分类 D_j 的准确率和召回率分别定义为：

$$P=\frac{N_{ij}}{N_{D_j}},R=\frac{N_{ij}}{N_{C_i}} \tag{8}$$

其中，N_{ij} 为聚类簇 C_i 中包含预定义分类 D_j 中元素的个数，N_{D_j} 为预定义分类 D_j 中元素的个数，N_{C_i} 为聚类簇 C_i 中元素的个数。对于预定义分类 D_j，F1-measure 值最高的聚类簇被认为是 D_j 的最佳映射。最终网页集合的整体聚类效果评估如下：

$$F=\frac{\sum_{j=1}^{j}(num(D_j)\times F(j))}{\sum_{j=1}^{j}num(D_j)} \tag{9}$$

其中，$num(D_j)$ 代表预定义分类 D_j 中元素的个数。F 值越高，表示聚类的效果越好。

5 实验与分析

5.1 实验硬件环境

为了验证改进后的树路径模型的有效性和可行性，本文对传统的树路径模型[6]和改进后的树路径模型进行了对比实验。实验平台为 PC(Intel Core i7、CPU 2.3GHz、内存 6GB)，算法使用 java 语言实现，绘图使用 R 语言绘制，所使用的测试数据集来自 CCF 科研数据中 2006 年 3 月中文网页分类训练集 CCT2006。

中文网页分类训练集 CCT2006 是根据常见的新闻类别而设定的分类体系，从新闻网站上抓取得到对应类别的新闻网页作为训练集页面。它包括 960 个训练网页和 240 个测试网页，分布在 8 个类别中。

5.2 相似度计算时间和结果统计

本文首先选取两篇腾讯新闻网页 A_1、A_2 和一篇新浪新闻网页 B 作为实验对象，分别使用传统模型和改进后的模型计算各网页之间的相似度。表 4 给出了统计结果。

表 4　传统模型与改进模型计算时间对比

网页对象	A_1A_2	A_1B	A_2B
传统模型计算时间	1133ms	827ms	718ms
改进模型计算时间	1287ms	1046ms	831ms
传统模型计算结果	0.965	0.843	0.812
改进模型计算结果	0.947	0.733	0.682

从表 4 中相似度计算时间可以看出，改进后的树路径模型在相似度的计算效率上要略低于传统树路径模型，这是因为改进后的树路径模型中最佳匹配串的搜索方法使用的是不完全匹配，相比于传统树路径模型的完全匹配，相似度的计算结果更加准确、合理。由于网页文档中的树路径规模基本都是较小的，因此两种方法在计算时间上基本没有太大差距。从实际情况考虑，树路径规模巨型的文档数据也较为罕见，因此改进后的树路径模型适用于大部分情况。综合聚类效果图和相似度计算结果来看，改进后的树路径模型从 3 个因素考虑了网页结构的相似度，不仅对结构差别较大的网页进行了良好的区分，而且较好地反映了来自同一模板的网页之间的差异性。

5.3　聚类效果评估

为了验证改进后树路径模型的有效性和可行性，本文从训练集 CCT2006 3 个类别的新闻网页中各挑选 50 篇，并计算网页之间的相似度，建立相似度矩阵，采用凝聚层次聚类算法对网页集合进行聚类。使用传统树路径模型和改进后树路径模型的层次聚类的效果分别如图 5 和图 6 所示。

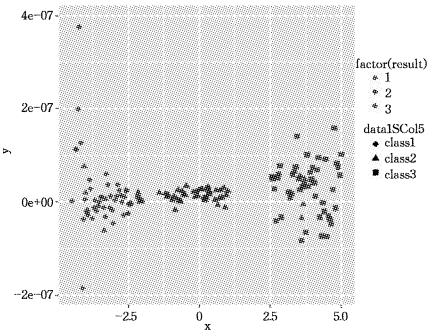

图 5　传统树路径模型层次聚类效果

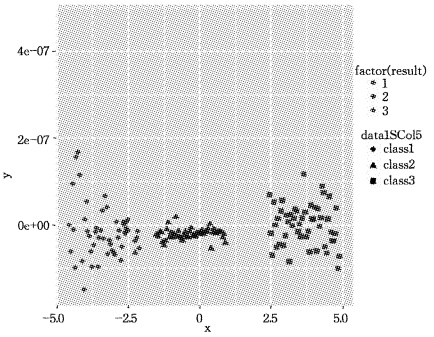

图 6　改进后树路径模型层次聚类效果

针对上述实验数据，传统树路径模型和改进后的树路径模型的各项聚类指标对比结果如表 5 所列。

表 5　传统模型与改进模型聚类指标的对比

指标	传统模型	改进模型
召回率	0.9293	0.9933
准确率	0.9355	0.9935
F 平均值	91.8%	99.0%
聚类个数	3	3
各簇数目	58,38,54	51,49,50
错误个数	12	1

在选取的实验数据集合中，从视觉上观察到有些网页之间部分模块较为相似，结合聚类结果可以推测出，传统树路径模型极可能在上述情况中出现路径重复匹配的现象，导致聚类结果产生了较高的错误率。然而改进后的树路径模型在聚类分析中取得了更优的效果，更加准确地度量了网页之间结构的相似度，提高了识别网页结构相似性的能力，弥补了传统树路径模型的不足。

结束语　DOM 树结构相似性的度量有着广泛的应用，如信息提取、模式抽取、组织结构的比较等。本文提出了一种改进的树路径模型，给出了 DOM 树结构的相似性计算方法。该方法简单高效、适用范围广，同时在网页聚类中比传统树路径模型取得了更好的聚类效果，提高了识别网页结构相似性的能力。由于改进后的树路径模型使用的是不完全匹配，造成计算效率上相比传统树路径模型有一定的下降。

为了进一步完善此模型并能高效地处理海量数据，未来的工作主要集中于并行计算或分布式架构的算法研究；此外考虑到不同的路径在相似度计算上发挥的作用不同，因此需要对路径进行选择，从而构造一个更加快速、准确的相似性度量方法。

参 考 文 献

[1] Li Yan-heng. The XML-based Information Extraction on Data-intensive Page[C]// IFIP International Conference onNetwork and Parallel Computing Workshops, 2007. NPC Workshops, IEEE, 2007: 1027-1030

[2] Li R, Pei C, Zheng J. Web Information Extraction Based on Hybrid Conditional Model[C]// 2010 Second International Workshop on Education Technology and Computer Science (ETCS). IEEE, 2010, 1: 137-140

[3] 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(z3)：1-6

[4] Tai K C. The tree-to-tree correction problem[J]. Journal of the ACM (JACM), 1979, 26(3): 422-433

[5] Cruz I F, Borisov S, Marks M A, et al. Measuring structural similarity among Web documents: preliminary results[M]// Electronic Publishing, Artistic Imaging, and Digital Typography. Springer Berlin Heidelberg, 1998: 513-524

[6] Joshi S, Agrawal N, Krishnapuram R, et al. A bag of paths model for measuring structural similarity in Web documents[C]// Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003: 577-582

[7] 王志琪，王永成. HTML 文件的文本信息预处理技术[J]. 计算机工程，2006，32(5)：46-48

[8] Gupta S, Kaiser G, Neistadt D, et al. DOM-based content extraction of HTML documents[C]// Proceedings of the 12th International Conference on World Wide Web. ACM, 2003: 207-214

[9] Bajcsy P, Ahuja N. Location-and density-based hierarchical clustering using similarity analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(9): 1011-1015

[10] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques (Third Edition)[M]. Thailand: Elsevier Pte Ltd, 2012: 297-302

[11] McCarthy J F, Lehnert W G. Using decision trees for conference resolution[C]// The Fourteenth International Joint Conference on Artificial Intelligence. 1995: 109-114