2016.5

李亮城

11521010

liangcheng\_li@zju.edu.cn

**结构化大规模网站聚类的高效算法**

（论文理解）

目录

[表目录 2](#_Toc452470470)

[1 背景与意义 3](#_Toc452470471)

[2 算法模型定义 3](#_Toc452470472)

[3 算法原理 4](#_Toc452470473)

[3.1 局部最优化 4](#_Toc452470474)

[3.2 小候选项效果 4](#_Toc452470475)

[3.3 大划分组效果 5](#_Toc452470476)

[4 算法实现步骤 5](#_Toc452470477)

[4.1 获取URL列表 5](#_Toc452470478)

[4.2 URL预处理 5](#_Toc452470479)

[4.3 MDL聚类 5](#_Toc452470480)

[5 实验结果分析 7](#_Toc452470481)

[6 小结 7](#_Toc452470482)

表目录

[表 4‑1　递归聚类算法伪代码 6](#_Toc452470242)

[表 4‑3　最优划分算法伪代码 6](#_Toc452470243)

结构化大规模网站聚类的高效算法

**李亮城 11521010**

# 背景与意义

任何从数据库中生成内容的网站都是通过一个或多个脚本来创建网页的，使得一个网站可以看做多个聚类，每个类都有一个脚本创建。由于大量的层次较浅的和较深的网站都是通过数据库来生成网页内容的，例如，购物网站，娱乐网站，学术存储库和图书馆目录，这些网站同时又是信息提取的常用对象。由同一个脚本产生的网页具有相似的结构使得信息提取系统可以用简单的规则来高效的提取信息，这样的系统称为 wrappers. wrappers 系统在商业上有广泛的运用,但是wrapper技术最初的目标和最重要的应用是结构化数据库的设计与应用，我们的研究目标会在这之上，针对精炼的网站链接数据产品和网站概念。其中关键的部分是在用户兴趣域中进行网站大规模信息提取的需求。

我们和传统的wrapper 系统的区别在于：输入格式的不同。传统的wrapper系统，给定一些训练数据作为输入，然后每个脚本对应的网页作为输出。对于我们的大规模网站提取，大量的网站作为输入，每个网站会得到不同数量的脚本作为输出。因此我们方法的最根本不同在于没有每个网站的训练数据。在已有的研究现状下，目前存在的问题就是：现有的技术不适用于大型网站。由于数据库产生网页的网站一般包含百万级别的网页，对它进行网页聚类来提取信息，我们希望可以在适当的时间内完成。现有的大部分信息提取技术的耗时都是巨大的。本文提出了一个可以适用于大型网站的聚类技术，仅仅依赖URLs和少量的内容特征。以前有过研究表明利用URL的相似性不能得到很好的网页聚类效果，但我们是从另外一个不同的方面使用URL。因此我们提出了一种原则性较强的框架，基于信息理论的原则，处理一系列为URL和内容提供最简单描述的脚本。

# 算法模型定义

考虑到时效性，URL聚类算法无法使用DOM树相关的一些网页特性，能使用的只有URL。观察一系列的网页URL，本文发现：网页的URL在绝大多少情况下包含了丰富信息，根据这些信息，可以对网页进行聚类。考虑下列URL：

URL1：http://www.giant.com.cn/Bike/BikeView/1208

URL2：http://www.giant.com.cn/Bike/BikeView/1166

URL3：http://www.giant.com.cn/News/NewsContent/377?ClassId=97

URL4：http://www.giant.com.cn/News/NewsContent/378?ClassId=97

前两个URL和后两个URL分别来自于两个不同的网页模板，可以看出共享网页模板的网页也同时共享一部分URL，URL根据“/”分割成若干个字符串，这里将分割生成的字符串称为一个URL项。其中“BikeView”和“NewsContent”很可能是服务器后台对应的一个模板，将这类URL项称为模板项，其他的URL项称为数据项。观察某个网站的所有网页以及URL，可以发现网站不同的模板项数量较小，而数据项数量较大。如果URL集合中的某个位置不同的URL项总数较小或者URL集合中的大部分网页都包含某个URL项，这些URL项很可能模板项，那么在划分网页集合时最好将它们分开。

对于某个URL：www.cdpf.org.cn/ggtz/content/2013-12/17/content\_30453768.htm 将其转化成如下形式：www.cdpf.org.cn/\*/content/\*/\*，其中“www.cdpf.org.cn”和“content”是模板项，剩下的3个URL项是数据项，因此称这个URL的度（数据项个数）为3。通过这样的形式，可以对网站所有网页URL进行编码，具体如下:

假设S是一个URL集合，为的所有URL项集合，S(t)表示集合S中包含某URL项t的URL集合，template(S)表示出现在S的每个URL中的URL项集合，即S的模板项个数，表示为。假设是集合S的一个聚类划分，其中每个聚类簇大小为，N为S的大小，用表示的数据项个数，即不是出现在所有的URL中的URL项。根据信息论的知识，对集合S的URL进行编码的模型如公式（1.1）所示：

公式 (1.1)

其中，为系数，所有模板项的长度为，聚类划分的索引编码长度为-，编码集合S的长度为。因为，其中表示中模板项的数量，可以转化为，系数和常数项、可以忽略，因此公式 (3.1)可以转化为：

公式 (1.2)

根据最小描述距离原则（MDL），选择使得最小的模型是最优的模型。

# 算法原理

## 局部最优化

假设为URL集合S的最优聚类模型，如果对于任意的，都有，表示W和S的交集。局部最优化引理意义重大，如果有两个不在同一个聚类簇中的两个URL集合，引理证明可以独立地分别寻找这两个URL集合的最优聚类划分组成最终的最优聚类划分。

## 小候选项效果

假设是一个URL项集合，对于如下聚类划分，并且，那么。用形式化的语言描述为：对于在给定位置URL项不超过的URL集合，按这些URL项划分集合得到的结果要比不划分得到的结果好。

## 大划分组效果

给定一个URL项t，假设为集合S中包含URL项t的URL比例，假设是t在S上的一个聚类划分，则总存在一个阈值，如果，那么。其形式化语言的描述为：如果一个URL项在集合中出现的比例达到一定阈值，那么按这个URL项划分URL集合比不划分效果要好。

对于这三个引理，文献[19]都给出了详细的证明过程，因此本文不再赘述其证明过程。

# 算法实现步骤

## 获取URL列表

类似于传统的网络拓扑，依据网页之间的超链接关系，可以将网站中的所有网页看成一个有向图，图中的有向边表示网页间超链接的指向关系。网络爬虫从一系列根节点出发，可以使用深度优先搜索或广度优先搜索策略遍历整个网站的网络拓扑获得网站的URL列表。

## URL预处理

根据URL中的“/”分割符将URL分解成若干项，因为出现的URL不同位置的同名URL项可能为两个不同的模板，因此在分解URL项时附加上其在URL中的位置信息，考虑这个URL：“site.com/a/b/…”，分解之后得到的URL项集合为{(POS1=site.com),(POS2=a),(POS3=b),…}。再考虑下面这个URL：

www.giant.com.cn/News/NewsContent/377?ClassId=97

参数名称相同的网页也很可能来自同一个模板，因此在分解URL时同时考虑参数项及其值，因此上面这个URL分解之后得到的URL项集合为{(POS1= www.giant.com.cn},(POS2=News),(POS3=NewsContent),(POS4=377),(ClassId),(ClassId=97)}。

## MDL聚类

文献证明了MDL聚类问题是NP-complete问题，并且提出了一种基于贪心算法的启发式算法。算法主要包括三个步骤：1) 递归地对集合进行聚类划分；2) 候选划分集合计算；3) 计算并选择最优的划分。

在递归聚类阶段，使用局部最优化引理，对集合S进行递归的聚类，每次选择mdl值最小的那个划分作为局部最优划分，直到mdl值收敛，模型达到最佳模型。算法伪代码如表 4‑2所示：

表 4‑1　递归聚类算法伪代码

|  |
| --- |
| 算法 1：递归聚类算法RCLUSTER |
| **输入**：S，一个URL集合  **输出**：C，一个URL聚类划分  1:  2: partition(S) //S的最优划分  3: **if** is not null **then**  4: **for** **each** **do**  5:  6: end for  7: else  8: return {S}  9: end if |

MDL聚类是NP-complete问题，因此在寻找最优划分时，使用贪心的思想选择mdl值最小的划分作为局部最优划分。考虑到小候选项效果和大划分组效果，贪心算法分为两步，第一步依据集合S是否包含出现最频繁的URL项t分为，计算mdl值并保存结果；第二步，首先将URL项根据其出现频率排序得到，定义如下候选划分集合，其中表示包含URL项但是不包含URL项的URL集合。文献[19]证明只要k最多取到集合的聚类个数时，划分总能包含最优解，为了避免重复计算，算法将计算候选集C与原集合S之间的mdl差值，即：

公式 (3.3)

其中表示集合新增的模板项个数。使用hash技术，算法能够在线性时间内计算出所有候选划分的。最优划分算法伪代码如表 4‑3所示：

表 4‑3　最优划分算法伪代码

|  |
| --- |
| 算法 2：最优划分算法PARTITION |
| **输入**：S，一个URL集合  **输出**：，一个贪心的URL划分，如果mdl没有减小则为空集  1:  2: //候选划分集合  3: //两路贪心划分  4: **for** each **do**  5:  6:  7: end for  8: //k路贪心划分  9: ，表示为在URL集合中出现最频繁的URL项排序  10: **for** **each** **do**  11:  12:  13:  14:  15: end for  16: //找到使得mdl减小最多的划分  17:  18: **if** **then**  19: return  20: **else**  21: return  22: **end** **if** |

# 实验结果分析

正确率和召回率是反相关的关系，随着α的增大，实验的准确率逐渐提升而召回率则大幅下降。因为当α=0时，所有URL都在同一个聚类簇中，因此召回率为1，而准确率则偏低；当α=∞时，所有URL都在各自的聚类簇中，因此准确率为1，召回率偏低。

线性化的运行时间，算法的运行时间与URL数量几乎成线性关系，其时间复杂度为O(n)

# 小结

本文基于最小描述长度原则设计了URL聚类算法，对于大规模网页特征提取节省了时间上的开销，根据URL的相似性进行聚类，在算法的执行过程中数据方面的复杂度也有所降低。最优化划分算法保证了算法的线性时间复杂度，对实际应用的可行性奠定了基础。