第 32 卷 第 15 期 计 算 机 工 程 2006 年 8 月
Vol.32 № 15 Computer Engineering August 2006
·发展趋势/热点技术· 文章编号：1000—3428(2006)15—0004—03 文献标识码：A 中图分类号：TP391

Web 挖掘技术研究

张 蓉
（广东商学院信息科学学院，广州 510320）

摘 要：随着互联网的飞速发展，Web 挖掘技术已成为数据挖掘技术的一个研究热点。该文对 Web 挖掘的特点、方法进行了讨论，设计了一种快速有效的 Web 文档聚类方法，给出了实际测试结果，验证了 Web 挖掘技术的有效性。提出的 Web 挖掘技术有效地提高了该系统的协作能力。

关键词：Web 挖掘；日志文件；文档聚类；矢量空间模型；关联规则

Research on Technology of Web Mining

ZHANG Rong
(Department of Information Science, Guangdong Commercial College, Guangzhou 510320)

【Abstract】 As the rapid development of the Internet, the technology of Web mining is now a hotter research field of data mining. This paper discusses the characteristics and methods of Web mining, and gives a fast and efficient Web document clustering method. It gives the experiment, which shows that the technology of Web mining is practical and efficient. The technology is suitable for the system and affords more guarantee and reliability.

【Key words】 Web mining; Log file; Document clustering; Vector space model(VSM); Association rule

随着 Internet 的迅猛发展，信息容量呈爆炸性增长趋势，然而信息检索工具和分析工具的相对落后，导致了信息过载。目前，人们从 Web 上获取信息的主要途径是通过搜索引擎，搜索引擎虽然部分地解决了资源发现问题，但其精度不高，不能为用户提供结构化信息，也不能提供文档分类、过滤等功能。因此，人们迫切需要能够从 Web 上快速、准确、有效地获取所需资源和有用模式的方法和技术，Web 挖掘技术便应运而生，并引起人们的极大兴趣。所谓 Web 挖掘是指从大量的数据集合 C 中发现隐含的模式 p。如果将 C 看作输入，将 p 看作输出，那么 Web 挖掘的过程就是从输入到输出的一个映射ξC→p 。与传统数据挖掘的对象绝大部分是结构化的数据库相比，Web 挖掘的对象是大量异质的、分布的 Web 数据集。就其挖掘的内容而言，Web 挖掘可分为 Web 内容挖掘、Web 结构挖掘和 Web 使用记录挖掘。

1 Web 挖掘的特点

Web 上的数据具有非结构化、动态、不完全、混沌的特点和巨大、分层、多维的形式，Web 挖掘与传统的数据挖掘相比，有其自身特有的性质与要求，具体如下：

(1)Web 数据源具有很强的动态性。Internet 本身就是一个时刻动态更新和变化的系统。这就需要借鉴数据仓库的某些技术，以此保存 Web 上动态更新的数据；

(2)Web 数据大多是 html 格式且有关某个主题的信息杂乱地散布在 Web 站点的多个目录下。这样就需要有一个强大的搜索引擎通过查找关键词来定位超文本的位置，并且这些搜索到的数据还可能存在数据冗余、不一致甚至矛盾等现象；

(3)Web 数据具有多样性。Web 数据经过滤后，既有数位型(整型、实型)、布尔型，又有分类数据、性质描述数据以及 Web 特有的数据类型如 url 地址、E-mail 地址等。这些新的数据类型必然带来新的特色，需要对原有数据挖掘方法进行改进和扩充；

(4)用户目标的模糊性。基于 Internet 的数据挖掘用户往往只对要挖掘的主题有一个粗浅的认识，提不出很明确的目标来，这就需要 Web 挖掘系统具有一定的智能性和学习机制，不断地跟踪用户的兴趣以挖掘出正确的结果，并清晰详尽地以用户能够理解的方式阐述出来；

(5)Web 数据目前以 TB 数量级计算，而且仍然在迅速增长，这就要求 Web 挖掘方法在对大数据集进行挖掘时依然具有高效率。

2 Web 挖掘方法

基于上述特点，Web 挖掘是一个极具挑战性的课题，它涉及 Web 存取模式、Web 结构和规则以及动态的 Web 内容查找。下面根据 Web 挖掘对象的不同，分类讨论 Web 挖掘的方法。

2.1 Web 内容挖掘方法

Web 内容挖掘是指从 Web 上的网页内容及其描述信息中获取潜在的、有价值的知识模式，以实现 Web 资源的自动检索，提高 Web 数据利用率的过程。它可以分为 Web 文本挖掘和 Web 多媒体挖掘。Web 文本挖掘是对 Web 上的大量文档集合的内容进行总结、分类、聚类和关联分析等。Web 多媒体挖掘是指从 Web 多媒体数据如音频、视频数据和图像等中抽取事先未知的、隐藏的、完整的和新颖的知识。由于当前 Web 上大多数信息的描述仍是以文本为主，因此目前关于 Web 内容挖掘的讨论主要是针对文本挖掘。

Web 内容挖掘的原理是先采用数据抽取和转换的方法将

基金项目：广东省自然科学基金资助项目(04009947)
作者简介：张 蓉(1966—)，女，副教授、硕士，主研方向：数据挖掘技术
收稿日期：2005-09-01 **E-mail：**zhangrong601@yahoo.com.cn

Web 上异质的、非结构化的 Web 信息进行整合和组织转换或映射为结构化的数据，再采用数据挖掘技术对其进行信息挖掘。本文最后所提供的一种快速有效的 Web 文档聚类方法正是采用了此原理。

2.2 Web 结构挖掘方法

Web 结构挖掘的对象是 Web 本身的超链接，即对 Web 文档的结构进行挖掘。对于给定的 Web 文档集合，通过算法发现它们之间链接的有用信息，文档之间的超链接反映了文档之间的包含、引用或者从属关系。

Web 结构挖掘在一定程度上得益于社会网络和引用分析的研究。把网页之间的关系分为 incoming 连接和 outgoing 连接，运用引用分析方法找到同一网站内部以及不同网站之间的连接关系。在 Web 结构挖掘领域最著名的算法是 HITS 算法和 PageRank 算法。它们的共同点是使用一定方法计算 Web 页面之间超链接的质量，从而得到页面的权重。著名的 Clever 和 Google 搜索引擎就采用了该类算法。

Page-Rank 算法的基本思想是：如果一个页面被多次引用，显然该页面很可能是重要的；一个页面未被多次引用但被一个重要页面所引用，该页面也很可能是重要的。一个页面中指向其它页面的超链接越多，在一定程度上说明该网页中的信息内容越有说服力，指向该页面的超链接越多即被多次引用，则说明该页面中的内容具有一定的权威性。因此，网页之间的超链接引用在一定程度上能表明 Web 文档的重要性；HITS 算法是利用 Hub/Authority 方法的搜索算法。它输出一组具有较大 Hub 权重的页面和具有较大 Authority 权重的页面。

2.3 Web 使用记录挖掘方法

在 Web 使用记录挖掘中，最主要的是 Web 服务器日志挖掘。Web 日志挖掘就是通过对网站日志文件的分析，获取网站访问情况的详细统计数据，针对这些统计数据进行数据挖掘，可以为网站管理者提供有用的知识，例如：针对每个用户的浏览行为进行挖掘，可以发现用户的访问模式，据此为用户提供个性化服务，从而提高网站的服务质量。

Web 日志挖掘流程可分为源数据收集、数据预处理、数据挖掘和模式分析 4 个阶段。例如 WebSIFT 信息过滤系统就是从 Web 站点中利用内容和结构信息挖掘出有用模式。(1)利用 Web 站点中的内容和结构信息，特别是 Web 服务器日志文件中的信息进行数据预处理，创造一个服务器会话文件，把服务器会话转换为事件，为数据挖掘阶段作好准备；(2)对服务器会话或事件文件进行序列模式分析、关联规则发现、聚类等；(3)利用简单的知识查询机制、可视化工具或信息过滤器对其结果进行进一步整理形成用户需要的最终知识。

3 一种快速有效的 Web 文挡聚类方法

快速有效的 Web 文档聚类方法的主要思想是：

(1)用VSM模型(Vector Space Model，VSM)[1]表示主题，建立文档——主题关联度二维表；

(2)利用关联规则挖掘算法Apriori算法[2]开采出主题频集，并据此扫描数据库，形成初步文档类；

(3)引入文档相似度函数，利用一种基于快速分解模拟退火算法的数据聚类算法生成最终文档类。

下面针对以上 3 个步骤，依次详细叙述每一步骤的具体操作。

3.1 Web 文档的结构化表示

Web 文档本身是半结构化或无结构的，且缺乏机器可理解的语义，为使其易于被计算机处理，Web 挖掘最基本的前期工作就是 Web 文档的结构化表示，这里，首先用矢量空间模型 VSM 表示每一个主题，再根据建立的主题特征向量依此分别计算给定文档与这些主题间的关联度，建立文档——主题关联度二维表。下面对所用到的概念，定义如下:

定义 1 主题特征向量。设T是主题的集合，对于其中的每一个主题$T_j \in T$，用特征向量表示：

$$\overrightarrow{T_j} = [(k_{j,1}, \omega_{j,1}), (k_{j,2}, \omega_{j,2}), ..., (k_{j,i}, \omega_{j,i}), ..., (k_{j,l}, \omega_{j,l})]^T \quad (1)$$

其中，$k_{j,i}$代表主题T_j中的第i个特征词条；$\omega_{j,i}$为第i个特征词条$k_{j,i}$对应的权值，表示该特征词条在该主题中的重要程度，且$\Sigma \omega_{j,i}=1$，$1 \le i \le \iota$；$\iota = \| \overrightarrow{T_j} \|$，为$\overrightarrow{T_j}$即$T_j$中特征词条的个数，各个主题的ι不同，依自身情况而定。

定义 2 文档与主题的关联度。关联度表示文档和某一主题之间的关联程度。设D是文档的集合，其中每一个文档$D_i \in D$，文档D_i和主题T_j之间的关联度$\lambda_{i,j}$，可按下式计算：

$$\lambda_{i,j} = \sum_{k=1}^{l} \mu_{j,k}^{i} \quad (2)$$

$$\mu_{j,k}^{i} = \| \cup K_{j,k} \| / \| \cup D_i \|^{\times \omega_{\cdot,\cdot}}$$

其中，$\iota = \| \overrightarrow{T_j} \|$，为向量的长度；$\mu_{j,k}^{i}$ 表示文档D_i对主题T_j中第k个特征词条的贡献；$\| \cup K_{j,k} \|$是主题T_j的第k($1 \le k \le \iota$)个特征词条$K_{j,k}$ 在D_i中出现的频度；$\| \cup D_i \|$即$\overrightarrow{T_j}$对D_i中有效词的个数；$\omega_{j,k}$为第k个特征词条$k_{j,k}$在主题特征向量中的权值。

这里，关联度$\lambda_{i,j}$将文档与某一主题间的联系用一个数据项表示，降低了数据的维数，因为如果直接用向量$[\mu_{j,1}^{i}, \mu_{j,2}^{i}, ..., \mu_{j,k}^{i}, ..., \mu_{j,l}^{i}]$来表示文档与主题的关联度，一方面数据的维数太高，处理效率低下；另一方面主题特征向量的长度ι也各不相同，无法表示。

此时，将文档视为事务，而将主题看作事务项(若关联度不为零，则对应的事务项出现，否则不出现)，并建立文档——主题关联度n×m二维表，表示如下：

$$\begin{array}{c} \lambda_{1,1}, \lambda_{1,2}, ..., \lambda_{1,j}, ..., \lambda_{1,m} \\ \lambda_{2,2}, \lambda_{2,2}, ..., \lambda_{2,j}, ..., \lambda_{2,m} \\ \cdots \\ \lambda_{i,1}, \lambda_{i,2}, ..., \lambda_{i,j}, ..., \lambda_{i,m} \\ \cdots \\ \lambda_{n,1}, \lambda_{n,2}, ..., \lambda_{n,j}, ..., \lambda_{n,m} \end{array}$$

其中，n和m分别为文档和主题的数量，表的每一行对应于一个文档，每一列对应于一个主题，表中（i,j）位置的值$\lambda_{i,j}$为文档D_i和主题T_j之间的关联度，由式(2)计算得到，且 $1 \le i \le n$，$1 \le j \le m$。

3.2 根据关联规则形成初步文档类

由于 Apriori 算法仅适用于二进制数据，因此首先要把文档－主题关联度二维表转换为一个二进制表，规定（i,j）位置的值若为 1 则表示文档 i 与主题 j 之间的关联度不等于零，否则（i,j）位置的值为零。

根据文档——主题关联度二进制表，可以用 Apriori 算法开采出主题频集；再根据主题频集扫描数据库，形成初步文档类（这里设初步文档类为 v 类）。这是基于一种思想即如果某些主题（即事务项）经常一起出现在某些文档（即事务）中，那么对应的文档（即事务）自然也是相似的，也就是说，根据关联规则开采算法得到的频集，可以找到对应的事务集

(即文档集)，并将它作为文档初步分类的结果。

尽管关联规则开采算法的计算复杂度会随着事物(文档)数目的增加而线性增长[3]，但如果选择合适的支持度阈值，不但可以控制Apriori算法的计算复杂度，还可以有效地祛除噪声点。通过反复试验，选取支持度阈值为4%较为合理。

3.3 根据聚类算法生成最终文档类

在利用聚类算法对初步文档集进行聚类之前，我们首先引入以下两个概念。

定义 3 文档特征向量。设D是文档的集合，对于其中的每一个文档$D_i \in D$，用特征向量表示：

$$\overrightarrow{D_i}=[(k_{i,1},\omega_{i,1}),(k_{i,2},\omega_{i,2}),\ldots,(k_{i,j},\omega_{i,j}),\ldots,(k_{i,n},\omega_{i,n})]^D \quad (3)$$

其中，$K_{i,j}$为特征词条，$\omega_{i,j}$为$K_{i,j}$ 在D_i中的权值，为$K_{i,j}$在D_i中出现的频率，且$\Sigma\omega_{i,j}=1$，$1\le j\le n$；$n=\| \ \|$，$\overrightarrow{D_i}$文档特征词条的个数。

为了获得文档的特征词条，可以先将 Web 文档进行分词处理，利用停用词表将停用词从文档特征集中剔除，然后运用倒排文档频率(IDF)约简特征词条，保留文档集中倒排文档频率在一定范围内的特征词条作为文档特征集，选用归一化词频(TF)作为特征词条的权值，这样就得到了一组文档特征向量。

定义 4 文档向量相似度函数. 文档向量的相似程度可以采用向量之间夹角的余弦函数表示，计算公式如下：

$$sim(D_i,D_j)=\cos(D_i,D_j)=\frac{\sum_{k=1}^{n}\omega_{ik}\times\omega_{jk}}{\sqrt{\sum_{k=1}^{n}\omega_{ik}^2}\times\sqrt{\sum_{k=1}^{n}\omega_{jk}^2}} \quad (4)$$

其中，$(\omega_{i1},\ldots,\omega_{in})$为文档$D_i$的特征向量，$(\omega_{j1},\ldots,\omega_{jn})$为文档$D_j$的特征向量。函数返回值在[0,1]之间，数值越大，相似性越大。

此时，可以采用一种基于快速分解模拟退火算法的数据聚类算法[4]生成最终文档类，该算法的最大优点是在数据维数较高时，仍能保持高效率。

其核心思想是：(1)把一个求解数据聚类问题转换为一个图形分割寻优问题，(2)利用模拟退火算法的原理求出优化问题的全局最优解。具体方法如下：

(1)把一个文档集表示为一个图形 G=（D，A），其中，

1)代表文档点集；

2）A 为连接两个相关文档端点的边集；

3)ω_j是对应于A中的每一条边a_j的权重($a_j\in A$)，它用来衡量由边连接起来的两个端点文档之间的相似程度，可由式(4)得到；

4)初步文档类v表示存在一个分割N把图形G大致分割为v个不相交的子图形$N_1,N_2,\ldots,N_v$；

5)费用函数$f=\Sigma\omega_j$，其中ω_j为特定边的权重($a_j\in A$)，连接任意 2 个不同的子图形Ni和N_i'中的点。($1\le i,i'\le v,i\ne i'$)；

(2)利用模拟退火算法的原理，寻找一种对图形G的分割，使得费用函数$f=\Sigma\omega_j$的值最小，即寻找一种分割使得各个子图形之间的相似性之和最小，即得到最终文档类。基于快速分解模拟退火算法的数据聚类算法见文献[4]。

3.4 实验证明

为了评价本算法，将它和K-Means算法[5]进行比较。实验中采用的Web文档集是用搜索引擎Yahoo从Internet上搜索得到的。整个实验在windows2000 平台上进行。

(1)测试算法的准确性。用 Yahoo 根据不同的主题进行了 40 次搜索，下载每次搜索到的前 20 个文档构成由 Yahoo 产生的 40 个文档类，共有 800 个文档；再用人工方法剔除无关文档，同样将它们分成 40 个类，并依此作为分类准确性的基准；然后分别采用本文算法和 K—Means 算法对这个文档集进行聚类。

由于不同的聚类算法产生的类的数目很可能不同，这里选用各自质量最好的 30 个类进行精度比较，结果由于本文算法采用二次聚类技术，因此平均聚类精度可达到 70%，而 K-Means 算法只能达到 40%。

(2)测试算法的的效率，同样用 Yahoo 在 Internet 上搜索前面 40 个主题的相关文档，但是下载的是每个主题的前 10 个文档，并以 10 的增幅逐步递增到前 40 个文档，形成文档数量依次为 400、800、1 200、1 600 的 4 个文档集；然后分别采用本文算法和 K-Means 算法对这 4 个文档集进行聚类，计算得到它们的平均聚类时间，如图 1 所示。由此可见，该算法在保持高精度的同时，其聚类速度也比 K—Means 算法要快。

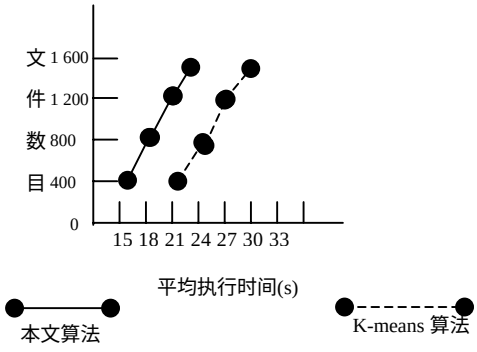

图 1 算法执行时间比较

4 结论

本文对 Web 挖掘的方法进行了讨论，并给出了一种快速有效的 Web 文档聚类方法，与传统的 K-Means 算法比较，本算法在聚类精度和效率方面都具有优越性。

参考文献

1 Salton G, Wong A, Yang C S. A Vector Space Model for Automatic Indexing[J]. Communications of the ACM, 1975, 18(5): 613-620.

2 Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules[C]. Proc. of the 20th VLDB Conference, Santiago, 1994: 487-499.

3 Cheeseman P, Stutz J. Advances in Knowledge Discovery and Data Mining[M]. The MIT Press, 1996: 307-328.

4 张 蓉，彭 宏. 一种基于快速分解模拟退火算法的数据聚类算法[J]. 计算机工程, 2002, 28(8): 88-89.

5 Bradley P S, Fayyad U, Reina C. Scaling Clustering Algorithms to Large Databases[C]. Proc. of the 4th International Conf. on Knowledge Discovery and Data Mining. AAAI Press, 1998-08.