基于句子级最大频繁单词集的 Web 文档聚类研究

路松峰　陈云开　袁　莉
（华中科技大学计算机科学与技术学院　武汉 430074）

摘　要　Web 文档聚类是 Web 挖掘的一个重要研究方向。现有的挖掘算法得到的频繁模式不仅维数高，而且不能很好反映文档表达的语义信息。为了得到更精确的聚类结果，本文提出一种基于句子级的最大频繁单词集挖掘方法来挖掘文档特征项。在此基础上，先初步聚类后依据类间距离和类内链接强度阈值合并或拆分类，最终实现文档聚类。在此过程中，使用可变精度粗糙集模型计算每个类的特征向量。实验结果表明，本文提出的算法优于传统的文档聚类算法。

关键词　Web 文档聚类，粗糙集，关联规则，最大频繁单词集

Research on Web Document Clustering Based on Sentential Maximum Frequent Word Sets

LU Song-Feng　CHEN Yun-Kai　YUAN Li
(School of Computer Science and Technology, Huazhong University of Science&Technology, Wuhan 430074)

Abstract　Web document clustering is an important research direction in Web mining area. Frequent pattern acquired form existing mining algorithms not only hashigh dimension, but can't reflects semantic information expressed form document well. For gaining more precise clustering result, this paper presents a mining algorithm based on sentential maximum frequent words set to mine document characteristic items. Based on then, documents are clustered elementarily at first. Then classes are incorporated or separated according to distance between classes and join intension in class. At the end, documents clustering is achieved. Variable precision rough set model is used to compute eigenvector of each class. The experiment results indicate the algorithm presented in this paper is better than traditional document clustering algorithms.

Keywords　Web document cluster, Rough set, Association rules, Maximum frequent words set

1　引言

随着 Internet 的快速发展，互联网上出现了海量的、异质的 Web 信息资源，其中 Web 文本信息占了主要地位。如何从浩瀚的 Web 文本信息资源中准确获取所需信息，已成为一个关键问题。信息处理日益成为当前最重要的研究内容，其内容包涵信息抽取、自然语言理解、自动聚类和分类、自动摘要、自动标注和主题识别、信息结构分析以及文本生成等。其中，关于自动聚类和分类的研究较为深入，并成为信息处理的核心技术[1]。

Web 文档聚类可以有效地压缩搜索空间，加速检索速度，提高查询精度。因此，文档聚类是 Web 挖掘的一个重要研究方向。文档聚类与分类的不同之处在于，聚类是一种无监督的文档分类，聚类没有预先定义好的主题类别，它的目标是将文档集合分成若干个簇，要求同一簇内文档内容的相似度尽可能大，而不同簇间的相似度尽可能小。文档聚类是搜索引擎的关键技术，Hearst 等人的研究已经证明了“聚类假设”，即与用户查询相关的文档通常会聚类得比较靠近，而与用户查询不相关的文档则相隔得比较远[2]。因此，可以利用文档聚类技术将搜索引擎的检索结果划分为若干个簇，用户便可以在自己感兴趣的簇中查看结果，或者根据聚类情况提出更精确的查询。

本文提出一种基于句子级的最大频繁单词集挖掘方法来挖掘文档特征项，使之能更有效地代表文档。在此基础上，建立文档集特征项矩阵，并计算其权值，通过计算向量点积对文档集进行初步聚类，然后依据类间距离和类内连接强度阈值合并或拆分类，最终实现文档聚类。

2　相关研究

从上世纪 60 年代以来，文档聚类算法一直是信息检索领域的一个重要的研究课题，而且已经取得了很多研究成果。这些成果已经被成功地应用到 Web 搜索引擎上，例如 Yahoo! Alta-vista 等。

目前，有多种文本聚类算法，大致可以分为两种类型：以 G-HAC 等算法为代表的层次凝聚法[3]，以 k-means 等算法为代表的平面划分法[4]。文[5]介绍了将 G-HAC 和 k-means 集合起来的 Buckshot 方法和 Fractionation 方法。在层次聚类法的研究领域中，JIAN-SUO XU[6]提出一种结合 LSA 和分等级 TGSOM 的 TCBLHT 方法。此方法能有效地解决传统的文本聚类算法所不能解决的分等级聚类问题。

聚类算法的研究已经十分成熟，但是由于文本的特殊结构，现有的聚类算法无法直接应用于其上。现有的文档聚类算法大部分是将文档表示成一种数学模型，然后对模型进行数学运算，求得模型代表的文档间的相似性，以此作为文档聚类的依据。因此，如何将文档表示成最能够代表自身的模型成为了文档聚类的关键，也是目前研究的热点。例如：S. T.

路松峰　博士，副教授，研究方向为数据挖掘、计算机网络；陈云开　博士研究生，研究方向为数据挖掘。

Dumais 等人提出的 LSA 模型[7]能够有效地表示文档间的语义联系。但是现有的模型不能很有效地代表文档，而且维数太高，给聚类带来太大的难度。本文提出的基于句子级的最大频繁单词集挖掘方法可有效地解决该问题。

3 建立文档集特征矩阵

3.1 文档表示模型

计算机没有类似人类的智能，人阅读完文章后可以产生自身对文章的理解，而计算机却没有这样的能力。为了便于计算机的处理，文档必须表示为计算机可以识别的格式。

文档聚类首先需对文本信息进行预处理，预处理技术主要包括分词(中文)、特征表示和特征提取等。文档中存在很多非结构化信息，以一定特征项(如词条或描述)来代表文档，在文档挖掘时只需对这些特征项进行处理，从而实现对非结构化文档的处理，这是从非结构化向结构化转化的处理步骤。

文档的表示模型有多种：布尔逻辑型、向量空间型(VSM)、概率型以及混和型等。本文采用向量空间模型对文本进行表示，向量空间模型将文本表示成特征项和特征项权重组成的向量，从而可把聚类问题转化成向量计算问题。

给定一个包含 n 个特征项的文档 D，D 可以表示为：$\{t_1:w_1, t_2:w_2, \cdots, t_i:w_i, \cdots, t_n:w_n\}$，其中 t_i 为文本特征项，w_i 为特征项 t_i 的权值，$i\in[1,n]$；文档 D 的特征项的集合 $T_D=\{t_1, t_2, \cdots, t_i, \cdots, t_n\}$；文档 D 的特征向量 $W_D=[w_1, w_2, \cdots, w_i, \cdots, w_n]^T$。文档 D 和其特征项集合 T_D 的维数都为 n，即 $|D|=|T_D|=n$。对文档 D、特征项集合 T_D 和特征向量 W_D 的第 i 项亦可分别用 $D[i]$、$T_D[i]$ 和 $W_D[i]$ 表示，即 $D[i]=t_i:w_i$，$T_D[i]=t_i$，$W_D[i]=w_i$。

3.1.1 文档特征项的获取

文档特征项的确定是影响文档聚类质量和速度的重要环节。本文利用关联规则挖掘算法来挖掘最能体现文档的特征单词集。关联规则是 Rakesh Agrawal 等人于 1993 年提出的数据挖掘领域中的一个重要研究方向[8]，最初主要应用于零售业。

现有的将关联规则用于文本特征项挖掘的模型主要有 ARC-BC[9]和 LB[10]分类器中用到的特征项模型。这两种模型的共同点是：将每个单词看作一个项目，每篇文档看作一个交易，而包含有同一类主题的文档集合被看作一个数据库。在这样的模型中所挖掘出的文本特征项集包含的是在一篇文档中频繁同时出现的单词。

句子作为一个语法单位在句法上独立，并表达了某个基本的语义，而一篇文档的中心思想也是通过组织句子的基本语义来表达的。同时出现在一个句子中的单词之间存在着或多或少的语义联系，并且与在同一文档中几个句子中出现的单词相比，出现在同一句子中的单词集包含了更多的局部信息，并具有更多的语义相关性，因而更适合作为文档的特征项。

基于以上的分析，为了更好地挖掘出句子中包含的局部信息，本文提出文档数据库模型。在文档数据库模型中，每个单词被看作一个项目，每个句子被看作一个交易，而每个文档被看作是一个数据库。利用关联规则挖掘算法可以挖掘出以句子为交易的频繁单词集，把其作为文档的特征项，来进行聚类。但当文档数量很大时，其所包含的频繁单词集也将非常多，由于最大频繁单词集包含了所有的频繁单词集，且数量远远少于频繁单词集，为了减少计算规模，利用最大频繁单词集来代表文档特征项，再在此基础上对文档进行聚类。

3.1.2 文档数据库模型

将文档转化成文档数据库模型，需要对每个文档进行数据预处理。为了检测出文档中的句子，先使用一种句子边界检测算法[11]，该算法基于有限状态机的使用启发规则的词法分析器。97%以上的句子边界可以被准确地检测出来，最终的文档中包含了非常准确的句子分隔符。为了去除那些对聚类贡献很小的单词，根据停词表去掉在文档中那些没有什么意义的词，对于英文文档还需要使用 Porter Stemmer 算法[12]去掉词的后缀。经过这些工作，可以更有效地在文档中挖掘局部上下文信息。

3.1.3 挖掘句子级的最大频繁单词集

将每个文档映射成数据库后，使用关联规则最大频繁项目集挖掘算法(例如 Max-Miner[13])来挖掘在每个文档中出现的最大频繁单词集，并将这些单词集作为文档的特征。和传统的挖掘算法不同的是，本文的挖掘算法是建立在文档数据库模型上的。

定义 1 单词集 X 的支持度：用 $\sup_D(X)$ 表示，$\sup_D(X)=|X|/||D||$，其中 $|X|$ 是文档 D 中包含单词集 X 的句子的个数，$||D||$ 表示文档 D 中句子的总数。

定义 2 频繁单词集：不小于用户指定的最小支持度阈值(minsup)的单词集称为频繁单词集。若 $\sup_D(X)\geqslant$ minsup，则 X 为频繁单词集，否则 X 为非频繁单词集。

定义 3 最大频繁单词集：如果频繁单词集 X 的所有超集都不是频繁单词集，则称 X 为最大频繁单词集。例如：文档 D 包含 4 个单词 $\{A,B,C,D\}$，假定 $\{A,B\}$ 是频繁单词集，而它的所有超集 $\{A,B,C\}$、$\{A,B,D\}$、$\{A,B,C,D\}$ 都不是频繁单词集时，则 $\{A,B\}$ 是最大频繁单词集。最大频繁单词集的所有子集都是频繁单词集。把文档 D 的所包含的最大频繁单词集作为 D 的特征项，则 D 的所有最大频繁单词集的集合即为 D 的特征项集合 T_D。

算法遍历所有的文档，对每个文档实施最大频繁项目集挖掘算法，挖掘出的每个文档的所有最大频繁单词集，得到每个文档的特征项集合。

3.2 建立文档集特征项矩阵

挖掘出每个文档的最大频繁单词集后，必须得对文本特征向量进行规范化表示，并计算特征项的权值，也就是最大单词集表示文档的近似程度。

给定一个包含 N 个文档的集合 $\Omega=\{D_1, D_2, \cdots, D_N\}$，其特征项的集合记作 T_Ω，$T_\Omega=\bigcup_{i=1}^{N}T_{D_i}$。文档集合 Ω 所包含的全部最大频繁单词集的数量为 m，即 $|T_\Omega|=|\bigcup_{i=1}^{N}T_{D_i}|=m$。对于任意一个 $D_i\in\Omega$，需要对其规范化，把规范化后的文档记作 D'_i，其生成方式如下：

$$T_{D'_i}[k]=T_\Omega[k]$$

$$W_{D'_i}[k]=\begin{cases}W_D[p], & T_{D'_i}[k]\in T_{D_i} \text{ and } T_{D'_i}[k]=T_{D_i}[p]\\ 0, & T_{D'_i}[k]\neq T_{D_i}\end{cases}$$

其中，$1\leqslant i\leqslant N$，$1\leqslant k\leqslant m$，$1\leqslant p\leqslant|D_i|$。

规范化后的文档的集合记作 $\Omega=\{D'_1, D'_2, \cdots D'_N\}$。$\Omega$ 的特征项矩阵：

$$W_\Omega=[W_{D'_1}, W_{D'_2}, \cdots, W_{D'_N}]=[w_{ij}]_{m\times N} \quad (1)$$

其中 $1\leqslant i\leqslant m$，$1\leqslant j\leqslant N$。w_{ij} 表示第 i 项(单词集)属于第 j 个文档的权值。应用 TF. IDF 方法[15]得到 w_{ij} 的值：

$$w_{ij}=\frac{tf_{ij}\times\log_2(\frac{N}{n_i}+0.01)}{\sqrt{\sum_{k=1}^{m}(tf_{ik}\times\log_2(\frac{N}{n_i}+0.01))^2}} \quad (2)$$

其中 tf_{ij} 表示第 i 个单词集出现在第 j 个文档中的频度，$tf_{ij}=\sup_{D_j}(T_\Omega[i])\times\|D_j\|$。$n_i$ 表示包含第 i 个单词集的文档数，$n_i>0$。由公式(2)可知，$w_{ij}\geqslant 0$，当项目 i 在文档 j 中出现的频度为 0 时，$w_{ij}=0$。

特征项矩阵 W_Ω 可以看成是由 N 个文档特征列向量组成的。每一列代表着一个文档的特征向量。给定一个规范化文档 D_j，j 表示文档集 Ω 中的第 j 个文本，$D_j=\{t_{1j}:w_{1j},t_{2j}:w_{2j},\cdots,t_{ij}:w_{ij},\cdots,t_{mj}:w_{mj}\}$，$W_{D_j}=[w_{1j},w_{2j},\cdots,w_{ij},\cdots,w_{mj}]^T$，$W_{D_j}$ 是矩阵 W_Ω 中的第 j 个列向量。

4 Web 文档聚类

文档聚类的算法主要有基于概率和基于距离两种方法。基于概率的方法以贝叶斯概率理论为基础，用概率的分布方式描述聚类结果。基于距离的方法，如 k-means 和最近邻居等，都以传统的特征向量表示文档，再将文档看作是向量空间中的一个点，通过计算点之间的距离进行聚类。本文选用基于距离的方法。

对于大量的文本集，如果通过向量空间计算两两文档的相似性，当向量空间的维数很大时，将会耗费大量时间。本文先定义一个简单的判断函数将文本集预先聚类，然后依据类间耦合度和类的内聚性进行聚类确认，得到最理想的聚类结果。

4.1 初步聚类

给定文档集 Ω 和 Ω 中的任意两个规范化的文档 D_i 和 D_j，则定义它们的点积为：

$$\mu_{ij}=D_i\cdot D_j=\sum_{k=1}^{m}(W_{D_i}[k]\times W_{D_j}[k]) \quad (3)$$

μ_{ij} 表示两个向量的点积，其每一个分量按公式 $\mu_{ij}[k]=W_{D_i}[k]\times W_{D_j}[k]$ 计算，反映两个向量的近似程度。μ_{ij} 越大，说明两个向量具有相同的项越多，当 μ_{ij} 大于给定的阈值 *minu* 时，初步认定两个向量代表的文本是相似的。可以将相似的文本归为一类。而且，本文规定在此步骤中，聚类具有传递性。

定义 4 文档 D_i 和 D_j 相似记为 $D_i\approx D_j$，不相似记为 $D_i\not\approx D_j$，如果有 $D_i\approx D_j$，$D_i\approx D_k$，则 $D_j\approx D_k$，即 $D_i\in C,D_i\in C,D_k\in C$，$C$ 是 Ω 的一个子集，表示一个类别。

由于点积的计算速度很快，因此可以很快确定初步聚类，初步聚类的计算速度还和阈值的选择有关，如果阈值选择太大，算法速度很快，但会把所有的文档分到同一个类中；阈值选择得太小，则会把所有的文档分到不同的类中。太大或太小的阈值都会导致初步聚类不理想。

4.2 计算每个类的特征向量

通过初步聚类，可将文档集 Ω 表示成 $\Omega=\{C_1,C_2,\cdots,C_h\}$，$h$ 为初步聚类的类别数。但是初步聚类只是通过粗略的计算得到的，如果一个文档向量的权值太大，会影响其它的文档和它的相似性的判断结果。要得到精确的聚类结果，必须对初步聚类的结果进行更精确的聚类。在进行精确聚类以前，需要计算每个类的特征向量。

本文采用可变精度粗糙集模型[14]计算每个文档的最大频繁单词集对初步聚类的贡献，根据类中每个文档的频繁单词集的贡献大小计算类的特征向量。

设 (U,R) 为近似空间，其中，U 是所有研究对象的集合，称为论域，对应文档集 Ω，R 是定义在 U 上的等价关系。

定义 5 设 R 是定义在集合 A 上的一个关系，若 R 是自反的，对称的且传递的，则 R 称为等价关系。

定义 6 设 R 为集合 A 上的等价关系，对任意 $a\in A$，$[a]_R=\{x|x\in A,xRa\}$ 称为 a 关于 R 的等价类。其等价类集合称为 A 关于 R 的商集，记为 A/R。

容易证明定义 4 定义的相似关系为等价关系，记为 B。对于任意 $D_j\in C_i$，$D_k\in C_i$，$C_i\in\Omega$，有 $D_j\approx D_k$。因此 C_i 是 D_j 关于 B 的等价类。$U/B=\{C_1,C_2,\cdots,C_h\}$。

定义 7 假设 $X\subseteq U,Y\subseteq U,F(X,Y)=\begin{cases}|X\cap Y|/|X|, & |X|>0\\ 0, & |X|=0\end{cases}$，其中 $|X|$ 表示集合 X 的基数。$F(X,Y)$ 表示 X 关于 Y 的相对正确分类率，即如果将集合 X 中的元素分到集合 Y 中，则做出分类错误的比例为 $(1-F(X,Y))\times 100\%$，真正错分类的元素数目为 $F(X,Y)\times|X|$。

令 $\beta\in[0.5,1]$，$F(X,Y)>=\beta$ 表示 X 和 Y 中的公共元素的数目大于 X 中元素数目的 50%。显然，$X\subseteq Y\Leftrightarrow F(X,Y)=0$。

如果在初步聚类时，选取不同的阈值 *minu*，集合 U 可以定义另一个等价关系 E，也可定义另一个等价类集合 $U/E=\{E_1,E_2,\cdots,E_t\}$，$t$ 为类别数。根据可变精度粗糙集理论，定义 C_i 的 β 下近似为：

$$R_\beta C_i=\bigcup\{E\in U/E|F(E,C_i)>=\beta\} \quad (4)$$

$R_\beta C_i$ 也称为 C_i 的 β 正区域，可理解为将 U 中的元素以小于 $1-\beta$ 的分类误差分给 C_i 的集合。

C_i 中的某一文档在 $R_\beta C_i$ 中出现的频率越高，说明这个文档分类的正确率越高，文档的最大频繁单词集对分类的贡献越大。因此，C_i 中的文档在 $R_\beta C_i$ 中出现的频率能代表文档最大频繁单词集对分类的贡献，也就是最大频繁单词集的全局权值 γ。显然，$\gamma\in[0,1]$。

对于每一个类的特征向量计算如下：

$$W_{c_i}=\sum_{j=1}^{q}D_j\cdot\gamma_j \quad (5)$$

W_{c_i} 表示第 C_i 类文档的特征向量。q 表示 C_i 类中一共有 q 个文档。γ_j 表示 C_i 类中第 j 个文档的 γ 值。

4.3 精确聚类

对初步聚类得到的文档类进行确认分为两步(1)计算不同文档类之间的相似度，对类间相似度不小于指定阈值 minsim 的类进行合并。(2)计算文档类内的内聚度，对内聚度不大于指定阈值 maxcon 的类进行拆分。

给定包含 h 个类的文档的集合 $\Omega=\{C_1,C_2,\cdots,C_h\}$，对于任意两个类 C_i 和 C_j，它们之间的相似度 $\mathrm{sim}(C_i,C_j)$ 如下：

$$\mathrm{sim}(C_i,C_j)=\frac{\sum_{k=1}^{m}(W_{C_i}[k]\times W_{C_j}[k])}{\sqrt{\sum_{k=1}^{m}W_{C_i}[k]^2}\times\sqrt{\sum_{k=1}^{m}W_{C_j}[k]^2}}$$

其中，$C_i\in\Omega,C_j\in\Omega,1\leqslant i\leqslant h,1\leqslant j\leqslant h$。

类间相似度反映两个文档类的耦合程度，相似度的取值范围在 $[0,1]$ 之间，两个类越相似则相似度越大，反之则相似度越小。

给定一个包含 q 个文档的文档类 $C=\{D_1,D_2,\cdots,D_q\}$，对于任意一个文档 $D_i\in C$，定义其与类的内聚度 *con* (D_i,C) 如下：

$$con(D_i,C)=\gamma\sum_{k=1}^{q}(D_i\cdot D_k)=\gamma\sum_{k=1}^{q}\sum_{j=1}^{m}(W_{D_i}[j]\times W_{D_k}$$

[j])

内聚度实际上反映了某文档对其所属类贡献的大小，内聚度越大则表明该文档对类的贡献越大，反之则表明该文档对类的贡献比较小。贡献小的文档自然应该从所属类中去除。通过设置阈值可把对类贡献不大的文档从当前类中分出去。

5 算法描述

算法首先得到每个文档的特征项和支持度，然后对挖掘结果进行规范化，得到新的文档的特征项集合，并计算每个特征项的权值，为了降低算法的计算规模，先对文档集合进行初始聚类，对于结果要求比较粗糙的应用，此时算法即可结束。对于结果要求比较精确的应用，再在上述结果基础上进行精炼，根据类间相似度和类内内聚度，循环对聚类结果进行合并和拆分，直到得到稳定的输出。算法描述如下：

```
1)Ω = {D1, D2,…, DN}, num = 0
2)for each Di∈Ω do TDi = Max-Miner(Di)// 计算最大频繁单词集
3)规范化 Ω = {D'1, D'2,…,D'N},计算 each wij并生成特征项矩阵
4)for each (D'i∈Ω, D'i∈Ω)and D'i≠D'j do // 初步聚类Ω= {C1,
  C2,…, Ch}
5)    if μij >= minu then Unite(D'i,D'j,Ω)
6)end for
7)for each Ci∈Ω 计算其特征向量
8)do // 精确聚类
9)   num = |Ω|
10)   for each (Ci∈Ω, Cj∈Ω)and Ci≠Cj do
11)      if sim(Ci, Cj)>= minsim then Unite(Ci, Cj,Ω)
12)   end for
13)   for each Ci∈Ω do
14)      for each Dk∈Ci do
15)         if con(Dk, Ci)<= maxcon then Split(Ci, Dk,Ω)
16)      end for
17)   end for
18)while (num == |Ω|)
```

在算法中的函数 Max-Miner 为计算最大频繁项目集的经典算法，用来计算某个文档句子级的最大频繁单词集。函数 Unite 用来合并两个类或者两个文档，合并后将产生一个新的类，该类将包含原来的两个类中所有的文档，合并后特征项的权值为原来特征项权值的和，把合并后的类并入文档集合 Ω，并把原来的两个类从文档集合 Ω 中删除。函数 Split 则把文档 D_k 从类 C_i 中分出来，变成两个独立的类，把原来的类 C_i 从文档集合 Ω 中删除。

6 实验结果

聚类算法的速度以及聚类结果的正确性是衡量聚类效果的重要指标。以下的实验从这两个方面来衡量本文提出算法的优劣。文本提出算法包括挖掘文档最大频繁单词集和利用文档单词集聚类两个部分。本文设计两个实验分别测试这两个部分。实验是在赛扬 2.4G CPU、512M 内存、Windows XP SP2 操作系统的计算机上进行的。

实验 1：测试聚类部分的性能。给定 10 类文档，每类 5 个文档。手工设定文档的特征项并给出权值，表示成文档特征向量形式。本文的聚类算法与 k-means 和 TCBLHT 算法的比较如表 1，本文算法的参数 minsim=0.75，maxcon=0.15。

从实验结果可以看出，k-means 算法的准确率比较高，但是所花的时间也比较多，而且在聚类前必须指定聚类的类别数 k。TCBLHT 算法的主要优点在于能够得到分层的聚类结果，如果在不要求分层的情况下，此算法在时间和准确率上体现不出优越性。本文提出的算法因为有初步分类的预处理步骤，能够大大缩小聚类所花的时间，而且准确率也比较高。

实验 2：测试句子级最大频繁单词集作为文本特征项的性能。利用 Yahoo 根据不同的主题进行 50 次搜索，下载每次搜索到的前 20 个文档构成由 Yahoo 产生的 50 个文档类，一共包含了 1000 个文档。再用人工方法剔除了无关文档，并依次作为聚类准确性的基准。

使用 Max-Miner 算法挖掘文档的最大频繁单词集，使用 Apriori 算法[8]挖掘文档的频繁特征项。都使用 TF. IDF 方法计算特征项的权值，并建立文档向量。然后都使用 k-means 算法对两组文档特征向量集进行聚类，以便考察句子级最大频繁单词集作为文本特征项的性能。这里使用 k-means 算法是由于 k-means 比较稳定，类别数 k 取 50。

由图 1 和图 2 可以看出，基于文档挖掘出来的特征项和基于句子挖掘出来的特征项的数量都随最小支持度的增大而减少。当最小支持度相同时，本文算法得到的特征项数量要远小于文档级特征项的数量，其基于句子级的最大频繁单词集的聚类效果要比基于文档特征项的聚类效果好。由于句子级的最大频繁单词集数量少，用同一种聚类算法对两组特征项聚类，句子级的最大频繁单词集算法所花的时间要更少些。因此，句子级的最大单词集用较少的单词数量表达了更多的文档信息。

实验 1 表明本文提出的聚类算法同其它的聚类算法比，对于同样的文档特征向量，不仅时间短，而且聚类精度较高。实验 2 表明本文提出的基于句子级挖掘最大频繁单词集的思想能够得到数量少但能更能代表文档语义特征项。综上所述，本文提出的基于句子级的最大频繁单词集的聚类算法具有一定的优越性。

小结 Web 文档聚类在信息检索、自然语言处理和电子商务等领域有着广泛的应用。本文借用关联规则挖掘算法挖掘句子级的最大频繁单词集。将文档表示成空间向量模型，并用本文提出的算法对文档进行聚类，实验表明，本文提出的算法具有一定的优越性。

表 1 算法聚类部分的性能比较

算法	k-means(k=10)	TCBLHT(layer=2)	Our alg(minu=10)
时间(s)	30	22	8
准确率(%)	93	82	92

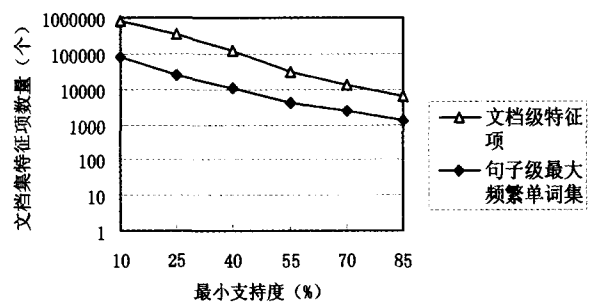

图 1 不同形式特征项数量随最小支持度的变化

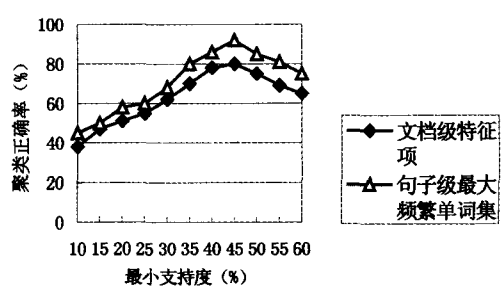

图 2 不同形式特征项的聚类准确率随最小支持度的变化

（下转第 164 页）

in chemical compounds. In:Proc. of the KDD-98, 1998
17 Mannila H, Toivonen H. Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery, 1997, 1(3):241～258
18 Ullman J D. Principles of Database and Knowledge-Base Systems. volume I. Rockville, MD: Computer Science Press, 1988
19 Nedellec C, Ade H, Bergadano F, et al. Declarative bias in ILP. In:De Raedt L, ed. Advances in Inductive Logic Programming, volume 32 of Frontiers in Artificial Intelligence and Applications, IOS Press, 1996. 82～103
20 Blockeel H, Raedt L D. Top-Down Induction of First-order Logical Decision Trees. Artificial Intelligence, 1998, 101(1-2): 285～297
21 Blockeel H, De Raedt L, Jacobs N. Scaling up ILP by learning from interpretations. Data Mining and Knowledge Discovery, 1999, 3(1):59～93
22 De Raedt L, Dzeroski S. First order jk-clausal theories are PAC-learnable. Artificial Intelligence, 1994, 70:375～392
23 Kietz J U, Lubbe M. An Efficient Subsumption Algorithm for Inductive Logic Programming. In: Proc. of the Eleventh Intl Conf. on Machine Learning, Morgan Kaufmann Publishers, Inc, San Mateo, CA, 1994. 130～138
24 Blockeel H, Dehaspe L, Demoen B, et al. Executing Query Packs in ILP. In: Proc. of the l0th Intl Conf. on Inductive Logic Programming, volume 1866 of Lecture Notes in Artificial Intelligence, Springer, London, UK, 2000. 60～77
25 Nijssen S, Kok J N. Faster Association Rules for Multiple Relations. In: Proc. of the 17th Intl Joint Conf. on Artificial Intelligence, Morgan Kaufmann Publishers, Inc, Seattle, USA, 2001. 891～896
26 Nijssen S, Kok J N. Efficient frequent query discovery in FARMER. In :13th Intl. Conf. on Inductive Logic Programming (ILP 2003), 2003
27 Clare A, Williams H E, Lester N M. Scalable Multi-Relational Association Mining. In:Proceedings of the 4th International Conference on Data Mining ICDM '04, 2004
28 Malerba D, Lisi F A. An ILP method for spatial association rule mining. In:Working notes of the First Workshop on Multi-Relational Data Mining, Freiburg, Germany, 2001
29 Lisi F A, Malerba D. Inducing Multi-Level Association Rules from Multiple Relations. Mach Learn, 2004, 55, 2 : 175～210
30 Han Jiawei, Fu Yongjian. Mining Multiple-level Association Rules in Large Databases. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(5)798～805
31 Schmidt-SchaubB M, Smolka G. Attributive concept descriptions with complements. Artificial Intelligence, 1991, 48(1):1～26
32 Buntine W. Generalized subsumption and its applications to induction and redundancy. Artificial Intelligence, 1988, 36(2):149～176
33 De Raedt L, Ramon J. Condensed Representations for Inductive Logic Programming. In: Proc. of Ninth Intl Conf. on the Principles of Knowledge Representation and Reasoning, 2004
34 Knobbe J, Blockeel H, Siebes A, et al. Multi-relational Data Mining. In:Proc. of Benelearn '99, 1999
35 Cook J, Holder H. Substructure discovery using minimum description length and background knowledge. J Artificial Intel Research, 1994, 1:231～255
36 Yoshida K, Motoda H, Indurkhya N. Graph-based induction as a unified learning framework. J of Applied Intel, 1994, 4:297～328
37 Inokuchi A, Washio T, Motoda H. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In: Proc. of the 4th European Conf on Principles of Data Mining and Knowledge Discovery, 2000. 13～23
38 Inokuchi A, Washio T, Motoda H. Complete Mining of Frequent Patterns from Graphs: Mining Graph Data. Machine Learning, 2003, 50: 321～354
39 Kuramochi M, Karypis G. Frequent Subgraph Discovery. In: Proc. of the 1st Intl Conf. on Data Mining, 2001
40 Kuramochi M, Karypis G. Discovering frequent geometric subgraphs. In:ICDM, 2002. 258～265
41 Yan X, Han J. gSpan: Graph-based Substructure Pattern Mining. In: Proc of the 2nd Intl Conf on Data Mining, 2002
42 Yan X, Han J. Closegraph: mining closed frequent graph patterns. In:KDD, 2003
43 De Raedt L, Kramer S. The levelwise version space algorithm and its application to molecular fragment finding. In:IJCAI'01: Seventeenth Intl Joint Conf on Artificial Intelligence. vol2, 2001. 853～859
44 Huan J, Wang W, Prins J. Efficient mining of frequent subgraphs in the presence of isomorphism. In:ICDM, 2003. 549～552
45 Nijssen S, Kok J N. A quickstart in frequent structuremining can make a difference. In:KDD, 2004. 647～652

(上接第 157 页)

参 考 文 献

1 王建会，申展，胡运发. 一种实用高效的聚类算法. 软件学报, 2004, 15(5): 697～705
2 Hearst M A, Pedersen J. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In: Proc. of the 19th Annual Int'lACM/SIGIR Conf. Zurich, 1996. 76～84
3 Willet P. Recent Trends in Hierarchic Document Clustering: A Critical Review. Information Processing and Mangement, 1988, 24(5): 577～597
4 Rocchio J J. Document Retrieval Systems—Optimization and Evaluation :[PhD dissertation]. Harvard University, Cambridge, MA, 1966
5 Cutting D R, Pedersen J O, Karger D R, et al. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections In: Proc. of the 15th Annual Int'l ACM/SIGIR Conf. Copenhagen, 1992. 318～329
6 Xu Jian Suo. Wang Li. TCBLHT: A New Method of Hierarchical Text Clustering. In: Proceedings of 4th International Conference on Machine Learning and Cybernetics, 2005. 2178～2181
7 Dumais ST, Furnas GW, Landauer TK, et al. Using Latent Semantic Analysis to Improve Information Retrieval. In: Proceedings of CHI'88, 1988. 281～285
8 Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, 1994. 487～499
9 Antonie M, Zaiane O R. Text Document Categorization by Term Association. In:Proc. of IEEE Intl. Conf. on Data Mining, 2002, 19～26
10 Meretakis D, Fragoudis D, Lu Hongjun, et al. Scalable Association-based Text Classification. In:Proc. of the 2000 ACM CIKM International Conference on Information and Knowledge Management 2000, 6～11
11 Aberdeen J, Burger J, Day D, et al. MITRE: description of the Alembic system used for MUC-6. In:Proceedings of the 6th Message Understanding Conference, 1993. 141～155
12 Porter M F. An algorithm for suffix stripping. Program, 1980, 14(3):130～137
13 Roberto J, Bayardo Jr. Efficiently Mining Long Patterns from Databases. In: Proceedings ACM SIGMOD International Conference on Management of Data, 1998. 85～93
14 Salton G, Buckley B. Term-Weighting approaches in automatic text retrieval. Information Processing and Management, 1988, 24(5): 513～523
15 Ziarko W. Variable precision rough set model. Journal of Computer and System Sciences, 1993, 46(1): 39～59