维普资讯 http://www.cqvip.com

SAT-TC：基于关联的层次文档聚类

李 曲 龙 昊
（华中科技大学计算机科学与技术系 武汉 430074）

摘 要 在一篇文档中，一个单词可以看作是一个项目，一组单词就是一个项目集。在以往的基于关联的文本聚类方法中，都是将一整篇文档看作是一个事务来挖掘频繁项目集和关联规则的。但是实际上，一篇文档中的基本语义单位是句子。在同一个句子中同时出现的一组词在语义上或多或少都是相互关联的，与分布在多个句子中的同一组词相比，前者要有意义得多。因此，基于以上发现，我们考虑将文档中的每一个句子看作是一个事务，一篇文档就被看作是一个事务的集合，并由此提出了一种新的文本聚类方法：SAT-TC。通过在标准测试集上的实验证明，SAT-TC 要优于传统的文本聚类算法。

关键词 关联规则，频繁项目集，句子关联事务，聚簇，层次文本聚类

Hierarchical Document Clustering Based on SAT Model

LI Qu LONG Hao
(Department of Computer Science Huazhong University of Science and Technology, Wuhan 430074)

Abstract A word in a document can be viewed as an item, and hence a group of words is an itemset. While previous association based clustering works all view a whole document as a single transaction for mining frequent itemsets and association rules, the basic semantic unit in a document is actually a sentence. Words co-occurring in one and the same sentence are usually associated in one way or the other, and are more meaningful than the same group of words spanning several sentences in the document. Based on the above intuitions, we present a novel document clustering method called SAT-TC where each sentence in a document is viewed as a transaction and the document is hence a set of transactions. The effectiveness of proposed SAT -TC method has been demonstrated better than traditional clustering algorithm by extensive experimental studies using popular benchmark text collections.

Keywords Association rule, Frequent itemset, Sentence as association transaction, Cluster, Hierarchical document clustering

1 引言

随着 Internet 和企业级 Intranet 的不断发展，越来越多的信息以电子文本的形式出现，其数量和内容比较庞杂，检索起来比较困难，因此如何自动、精确地去划分这些文档变得越来越重要。一种解决办法就是采用文本分类技术(Text Classification)。文本分类是一种有指导的机器学习方法，它需要专家手工确定一些带类标记的训练文本，然后利用这些训练文本进行学习以得到各个类的分类规则，最后使用这些分类规则对未标记的新文本进行自动分类。在没有训练文本集可用的情况下，我们需要采用另外一种解决方法：文本聚类技术。文本聚类也常常称为无指导的机器学习，其目标是将未知的文本集自动形成聚簇(cluster)，使得簇内的文本之间相似程度很高，而不同簇间的文本之间相似程度尽可能地低。

许多传统的聚类方法(比如 k-means[10]以及一些层次聚类算法[11])可以用于文本聚类，但都没有解决文本聚类所面临的一些特殊问题：很高的维度、大量的文本以及聚类完成后各个聚类描述的可读性。为了解决这些问题，最近提出了新颖的、基于频繁项目集(frequent itemset)的聚类算法。频繁项目集的概念源自于关联规则挖掘(association rule mining)[2]。一个频繁项目集实际上就是频繁出现在文档中的一组词，基于频繁项目集的聚类算法用它们来描述文档之间的相似性，并由此构建聚类。

据我们所知，目前所有的基于频繁项目集或基于关联的聚类算法都采用了以下的基本观点[5,7]：文本中的一个单词看作一个项目，每个文本表示成一条由这些单词组成的事务。

然而，文本的基本语义单元实际上是一条句子。同时出现在一条句子中的几个单词比分布在多条句子中的几个单词通常在语义上更加相互关联。比如说，假设有一篇文档有1000条句子，单词“wheat”出现在第一条句子中，我们可以比较下面两种情况：单词“farm”也出现在第一条句子之中；单词“farm”出现在第918条句子之中。显然，项目集{wheat, farm}在前一种情况下更有意义。

基于上面的观察，我们已经提出了一种新的文本模型——句子关联事务 SAT (Sentence as Association Transaction)，并把它应用于文本分类[1]。在 SAT 模型中，我们采用句子而不是文本作为关联挖掘的事务单元。在本文中，我们将 SAT 应用于文本聚类，从而提出一种新的基于关联的层次文本聚类算法 SAT-TC，它基于以下直觉：

1)反映“文档主题”的单词或者单词组合应该会在多条句子中出现，因为作者经常倾向于通过在不同的句子中重复那些单词来强调“句子主题”乃至“文档主题”。

2)在一个聚簇中，达到某个最小比例的不同文档应该共享一些反映相同文档主题的单词或者单词集合，也因此，这些

维普资讯 http://www.cqvip.com

共享的单词或单词集合某种程度上反映了"簇主题"。值得注意的是，一个簇的不同文档子集可能会共享不同的单词或单词集合，所有这些共享单词及单词集合就构成了这个簇的置信网络。

3)不同簇之间的文档应当尽可能少地共享单词或单词集合。

本文第2节给出SAT模型的基本定义以及利用它进行聚类的整体框架。第3节描述我们提出的基于SAT的聚类算法SAT-TC，实验及其评估则放在第4节，最后给出全文总结。

2 SAT的基本定义和整体框架

为了描述的方便，根据关联规则挖掘[2]中的一些概念，我们首先给出SAT模型的几个基本定义：

设$W=\{w_1,w_2,\cdots,w_n\}$是一组单词的集合，我们称$w_i(1\leqslant i\leqslant n)$为项(item)，$W$的任意非空子集(即包含至少一个项)，我们称之为项目集(itemset)。项目集中包含的项的个数称为项目集的长度，如W的长度为n。长度为k的项目集可记为k-ietmset。事务(transaction)定义为文本的一个句子。在文本中事务之间一般以句号、感叹号、问号、分号等句子终结符隔开。这样一个文本就成为了一个句子事务库。每个事务赋予一个唯一的事务号TID，TID=(docID,senID)，其中docID为文本号，senID为该事务在文本中对应的句子号。

如果项目集I是事务T的子集，即$I\subseteq T$，我们就称T包含I。如果在文本D中存在d个事务包含项目集I，那么项目集I在D中的支持数(support count)为d，支持度(support)定义为支持数与文本中总事务数的比值。如果在文本D中项目集I的支持数大于或等于用户指定的文本最小支持数，那么该项目集在文本D中就是文本频繁的，I称为文本D的文本频繁项目集(document frequent itemset)，文本D则称为I的支持文本。文本最小支持数的两个自然选择是1和2。最小支持数1是平凡的，可以称为基本文本最小支持度。直觉上，表达文本主题的单词应该出现在至少两个句子中，即重复出现，因此，我们称最小支持数2为自然文本最小支持数。

如果项目集I在整个文本集D的c个文本中是文本频繁的，那么I的全局文档支持度(global support)为$c/|D|$；如果文本频繁项目集I的全局支持度大于或等于用户指定的全局最小支持度(global minimum support)，那么项目集I称为全局频繁项目集(global frequent itemset)。

我们发现在Reuters-21578数据集中，即使小到只有3条句子的文本，也存在满足自然文本最小支持度的文本频繁项目集。例如文本No. 14012，它属于"TRADE"类，总共包含以下三个句子：

(1st) Federal Reserve Board Chairman Paul Volcker said reducing the federal budget deficit was a more important goal for Congress than drafting trade legislation.

(2nd) "Reduce the budget deficit," Volcker responded when asked by a member of the Senate Banking Committee about trade legislation priorities.

(3rd) "If you don't deal with the budget deficit, everything else you do is going to be counterproductive," he said.

通过对文本进行去停用词和word stemming后，不难发现上述文本包含以下5个最大文本频繁项目集(设频繁项目集的最大长度为5)，每一个频繁项目集的文本支持度都是2，也即百分值66.7%：

{volcker, reduc, trade, legisi, budget}

{volcker, reduc, trade, legisi, deficit}

{volcker, reduc, trade, budget, deficit}

{volcker, trade, legisi, budget, deficit}

{reduc, trade, legisi, budget, deficit}

注意，最大文本频繁项目集的任意子集一定也是文本频繁的，这也是挖掘频繁项目集的经典算法Apriori的基本思路[2]。例如：项目"reduc"是单词"reducing"的词干(stem)，它是一个文本频繁1-itemset，项目集{trade budget}则是一个文本频繁的2-itemset。另外，我们还可以看到，文本中诸如"said"、"the"等词也是文本频繁的，但是由于这些停用词对于文本分类聚类的意义不大，因此在文本的预处理过程中我们就将这些词除去了。

由此我们得到SAT模型的基本框架：

(1)将文本按照句子切分，每个文本转化为一个句子事务库；

(2)每条事务中的项用binary方式表示，即出现或者不出现；

(3)用包含某个项的事务数作为该项的支持数；

(4)文本用频繁项集集合描述。

我们的模型从空间上可以大大降低文本的维度，从语义上也较好地保持了文本的原义，而且避免了语义跨越现象。相对而言，我们的SAT模型更适合处理大文本，因为大文本更能满足我们的假设——反映文本内容的项目集至少会重复出现一次，这样就不会造成语义的丢失。

3 基于SAT模型的聚类算法

基于上述SAT模型框架，我们发展了一种新的基于频繁项集的聚类算法，这种算法有以下特点：

①聚类以频繁项目集为中心，每个簇中的文档都尽量共享某些核心的频繁项目集；

②扩展单个的中心频繁项集为聚类规则集，即多个频繁项集构成的规则集；

③聚类的层次性强。

3.1 构建初始聚类

对文本进行去停用词和word stemming后，根据句子划分文本，每个句子成为一条事务，对每个文本进行最小支持数为2(即自然文本最小支持数)的频繁项集挖掘，得到各个文本的文本频繁项集，文本则由这些频繁项目集构成的向量表示，即文档$d_i=(is_1,is_2,\cdots,is_n)$，其中$is_i(1\leqslant i\leqslant n)$为该文本通过将句子视为事务后挖掘出来的频繁项目集。

将所有的文本向量集合在一起构成了全局事务数据库G，其中G的每一条事务对应着一个文本向量，在对这个全局事务库进行频繁项集挖掘就得到了全局频繁项集集合。

以每个全局频繁项集为核心构造初始聚类，初始聚类的聚类规则即为该全局频繁项集，所有文档向量中包含该聚类规则的文档都归入该聚类。

3.2 构建层次关系

我们构造层次关系是依照聚类所含文档集的包含关系来确定的，但这种包含关系并不是完全的包含关系，而是一种近似的包含关系，这样做是为了同时考虑到聚类之间的相关性，使得生成的聚类更有意义。

定义1 给定两个聚类A、B和一个阈值$p(0\leqslant p\leqslant 1)$，如果$|A\cap B|/|A|\geqslant p$，则认为$A$是$B$的子类，记为$A\leqslant B$。

维普资讯 http://www.cqvip.com

另外，如果两个聚类很相似，则可以将其合并，减少生成的聚类的数量。

定义2 给定两个聚类 A、B，如果 $A \leqslant B$ 和 $B \leqslant A$ 同时成立，则认为聚类 A、B 相似，记为 $A \sim B$。

构建过程：

(1)构造一个根聚类 Root，所有的聚类都作为 Root 的子类；

(2)合并相似聚类：考虑每个聚类的子类集，对其中每个子类找到该集合中与之相似的聚类，并迭代找出与相似聚类相似的聚类，构成一个相似聚类集合。合并相似聚类集合为一个新聚类，其中新聚类的文档集是所有相似聚类的文档集的并集，新聚类的规则集是所有相似聚类的规则集的并集，新规则的子类集是所有相似聚类的子类的并集。

(3)构建聚类层次：考虑每个聚类的子类集 $S=\{S_1, S_2, \cdots, S_n\}$，对每个子类 S_i，在该子类集中找出所有 S_k 满足 $S_k \leqslant S_i (k \neq i)$，将这些聚类下移一层作为 S_i 的子类，S_i 的规则集变为原规则集并上新的子类的规则集，文档集变为原文档集并上新的子类的文档集。

(4)迭代进行(2)、(3)两步，直到整个层次树构建完成。

3.3 disjoin 聚类

到现在为止，我们生成的聚类之间的文档集还大量存在着重叠现象，聚类要求同层次的类尽量不重叠，因此需要 disjoin 聚类。

现在每个文档有可能同时属于几个聚类，从其中找出一个认为最好的聚类作为该文档最终归属的类，一篇文档分到某个聚类的程度通过计算分到该类中的得分来评估，在得分最大的聚类中保留该文档，从其他几个聚类的文档集中删除该文档，达到 disjoin 的目的。

评分函数如下：

$$Score(d,C)=\sum_{R_i \in C.RuleSet} TransNum(d,R_i)$$

其中 $RuleSet$ 为该聚类的规则集，$TransNum(d,R_i)$ 为统计在文档 d 中包含规则 R_i 的事务数。

4 实验评估

本节将给出一系列实验数据以及实验结果，并与评价最好的传统聚类算法 Bi-k-means[8]进行比较。

4.1 实验数据集

我们选用国际上通用的聚类算法实验数据集 Reuters21578进行测试，由于 Reuters21578中存在多分类现象，从中挑选了只属于一个类的文档作为测试对象，共包括8654篇文档，分布在65个类中，其中最小的类只包含1个文档，最大的类包含3725篇文档。

4.2 评估方法

我们采用评估聚类精度的普遍方法 F-Measure，给定一个生成的聚类 C 和一个专家分好的类 F，通过以下公式计算 F-Measure 值：

$$Recall(C,F)=\frac{|C \cap F|}{|F|} \quad (1)$$

$$Precision(C,F)=\frac{|C \cap F|}{|C|} \quad (2)$$

$$F\text{-}Measure(C,F)=\frac{2 * Recall(C,F) * Precision(C,F)}{Recall(C,F)+Precision(C,F)} \quad (3)$$

整体聚类的精度则由以下公式计算：

$$F\text{-}All(C)=\sum_{F_i \in F} \frac{|F_i|}{|D|} * \max_{C_j \in C}(F\text{-}Measure(C_j,F_i)) \quad (4)$$

4.3 实验结果

实验参数包括全局最小支持度的设定，判断聚类相似性的阈值 p 的设定。在 Reuters 数据集上进行的一组实验结果如表1。

表1

NO.	全局最小支持度	阈值 p	F-Measure
1	0.0225	0.75	0.342
2	0.035	0.7	0.358
3	0.040	0.7	0.344
4	0.0225	0.7	0.383
5	0.0225	0.8	0.327
6	0.0225	0.65	0.363
Average	0.353		

另外我们利用 CLUTO 2.1提供的聚类算法包[12]对 Bi k-means 进行了两组实验，Bi k-means 需要输入的参数是最终生成的聚类的个数，结果如表2所示。

表2

NO.	聚类个数	F-Measure
1	30	0.35
2	60	0.30
Average	0.325	

从结果上可以看出，我们的算法要优于 Bi k-means，另外，由于 Bi k-means 不是层次性的聚类算法，因此在聚类结构上语义性不强，而我们的算法则强调了这一点，方便结果的浏览和利用。

结论 本文在先前提出的一种新的文本模型—SAT 的基础上设计了一种新的文本聚类算法，SAT 模型以句子作为事务来看待，一个文本就相当于一个事务数据库。我们在此模型基础上发展了一种基于关联的层次聚类算法，利用频繁项集作为聚类规则集，由于频繁项集是基于 SAT 模型得到的，因此比普通的关联聚类算法所用的频繁项集更有语义。实验证明，我们的算法相对于传统的评价最好的文本聚类算法 Bi k-means 取得了更好的效果。

参 考 文 献

1 Feng Jianlin, Liu Huijun, Long Hao, Fang Qiong. SAT: Sentence As Association Transaction for Text Classification. Submitted for publication

2 Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules. In: Proc. of the 20th Very Large Data Bases, 1994. 487～499

3 Borgelt C, Kruse R. Induction of Association Rules: Apriori Implementation. Otto-von-Guericke-University of Magdeburg, Germany

4 Berkhin P. Survey Of Clustering Data Mining Techniques. Accrue Software, Inc

5 Beil F, Ester M, Xu X. Frequent term-based text clustering. In: Proc. of ACM SIGKDD, 2002

6 Yiu Man Lung, Mamoulis N. Frequent-Pattern based Iterative Projected Clustering. University of Hong Kong

7 Wang B K. Hierarchical Document Clustering Using Frequent Itemsets. In: Proc. of SIAM Intl. Conf. on Data Mining, 2003

8 Steinbach M, Karypis G. A Comparison of Document Clustering Techniques. In: Proc. of ACM SIGKDD Text Mining Workshop, 2000

9 The Reuters21578 collection. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

10 Dubes R C, Jain A K. Algorithms for Clustering Data. Prentice Hall, 1988

11 El-Hamdouchi, Willet P. Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval. The Computer Journal, 1989, 32(3)

12 Karypis G. Cluto 2.0 clustering toolkit, 2002. http://www-users.cs.umn.edu/? karypis/cluto/.