文本挖掘的时态文本关联规则算法研究

张春燕[1] 孟志青[1] 袁 沛[2]
(浙江工业大学经贸管理学院 杭州 310023)[1] (浙江大学机械工程学系 杭州 310007)[2]

摘 要 由于数据库的频繁更新,时态数据库隐藏了大量的未知信息,因此针对实时更新的数据库应产生相应的时态关联规则。虽然关联规则算法已经被深入广泛地研究,但在文本数据中时态关联规则算法的研究还不多见。在深入了解时态关联规则算法及其在文本数据中的研究价值后,以时态文本为对象进行了时态关联规则算法的研究,建立了时态文本数据的时间表示模型,提出了文本时态关联规则算法 SPFM,最后通过实验对算法进行了有效性验证,结果表明该算法是正确可行的。

关键词 文本,时态,关联规则,垂直数据,有效时间,浮动
中图法分类号 TP311 **文献标识码** A

Mining Algorithm for Temporal Text Association Rules in Text Mining

ZHANG Chun-yan[1] MENG Zhi-qing[1] YUAN Pei[2]
(College of Business and Administration, Zhejiang University of Technology, Hangzhou 310023, China)[1]
(Department of Mechanical Engineering, Zhejiang University, Hangzhou 310007, China)[2]

Abstract Due to the frequent updates of the database, temporal database has hidden a lot of unknown information. Thus, temporal association rules for updating database should be generated. Although the association rules algorithm has been intensively studied, temporal association rules algorithm for the text data is also unusual. In this paper, we studied temporal association rules algorithm for the text, and established the temporal model for text. Then we presented mining algorithm SPFM for temporal text association rules. Finally, we got an experiment to check the effectiveness of the algorithm.

Keywords Text, Temporal, Association rules, Vertical data, Effective time, Float

1 引言

数据挖掘如果能考虑到时间属性,那么将更有助于揭示事物本质。于是,带有时间属性的数据挖掘,即时态数据挖掘(TDM)也应运而生了。参照经典的数据挖掘定义,TDM 可定义为:TDM 是对观测到的时态数据进行分析,目的是发现未知的关系和以数据拥有者可以理解且对其有价值的新颖方式来总结时态数据[1]。

非形式的时态数据是随时间变化且不同时间状态相互关联的数据。Agrawal 等[2]率先发表时间序列相似搜索的研究论文后,TDM 开始引起学术界的关注。虽然时态数据挖掘已经有了很多研究成果[3],但是时态数据挖掘的研究对象基本都是直接的数值数据,而对文本数据的时态数据挖掘却很少有研究。

时态数据关联规则的算法现在也已经较多,比如为挖掘频繁时态项集,Tarek 等人提出了 ITARM 算法[4],这是第一次提出将数据库进行划分来得到二阶频繁项集;C. H. L 提出了 PPM 算法,但该算法对增量挖掘的速度太慢;Cheng. Y. Chang 等人提出了 SPF 算法,但在数据库不断更新时 SPF 算法还不能对其进行时态挖掘[5];Mr. Chen 等人在影像数据库中建立了时态关联规则模型,该算法优点是对基本的阈值有一个可调节机制[6];Byon 等人在时态关联规则挖掘中提出指数平滑过滤[7]。上述算法模型都还不能有效地处理时态文本数据库,原因有:1)这些算法计算时时间复杂度仍太高。所以运用这些算法来挖掘时态文本就显得相当困难,而且使大型数据库挖掘出来的关联规则数量不够精简;2)没有考虑每个独立文本项各自存在的有效时间;3)每个项目缺少一个合理的支持度数。以前在做挖掘时都把事务集 D 的大小作为支持度数的计数基础,但数据库在不断变化,所以需要规范化支持度数,其目的是让每个事务集都有一个最小支持度数。本文根据时态事件模型和序列模型以及 Apriori 原则,在快速更新算法思想上产生新的算法 SPFM。

SPFM 算法主要有 3 个特点:1)由于文本数据的维度一般都很大,因此通过算法预处理将文本数据转换成垂直数据格式,而且每个 k 频繁项集的 TID 集都携带了计算该支持度所需的完整信息,所以不再需要扫描数据库来确定项集的支持度,便可大大提高程序效率;2)对于给定的时间粒度,时态数据库是由多个时间粒度上的记录组成的,因此在挖掘时态

到稿日期:2012-09-20 返修日期:2013-01-12
张春燕(1987—),女,硕士生,主要研究方向为数据挖掘,E-mail:zhangchun732@163.com;孟志青(1962—),男,博士,教授,博士生导师,主要研究方向为数据挖掘与数据仓库、管理信息系统、最优化理论、信用风险;袁 沛(1988—),男,博士生,主要研究方向为计算机图形。

数据库的频繁项集时，本算法就通过更新不同时间粒度的支持度数来确定频繁项集，并判断频繁项集在时间粒度上的连续性；3）如2）所述，时态数据库是和时间粒度有关的，那么从时态数据库挖掘出的关联规则也应该是和时间粒度有关的，即存在“有效时间”，不同于以往给定一个有效时间的方法，本算法引入一种判断机制，使得发现的有效时间由频繁项集本身来决定，最终我们获得的是一组浮动的“有效时间”。

需要指出的是，本文研究成果不仅仅局限于医学病毒论文数据库，另外也可运用于警务挖掘、计算机病毒研究等。本文通过SPMF算法挖掘1970－2010这40年间医学病毒论文之间的关联规则，找到这40年里有关病毒研究的一些发展规律及病毒本身的发展规律或病毒与病毒之间的联系等关联规则。

2 时态文本处理

2.1 时态文本处理

在对文档进行特征提取之前，需要先对文本进行预处理，对英文而言需进行Stemming处理[8]，中文的情况则不同，因为中文词和词之间没有固定的间隔，需进行分词处理。本文的文本对象为英文，所以只需对文本进行Stemming处理。

对于本文研究的医学病毒论文数据库，文本预处理的具体内容如下：

①英文大写换小写（都以小写字母表示，方便文本识别）；②删除空白记录；③将论文信息中的摘要AB和标题TI合并（将标题与摘要合并可以提高关键词的比重，增加提取文本向量的精度）；④处理论文发表时间DP列，只保留年份数字，方便提取有效时间；⑤对于记录太多的库，适当拆分表格（否则在程序处理时会内存溢出）；⑥根据文本内容提取合适的停用词表，对文本内容进行去停用词处理（由于停用词提取的有限性，最终挖掘出的频繁项会包含少量无意义的内容，对于这个问题，本文的处理办法是先进行1-频繁项集的挖掘，由于1-频繁项的数目比未处理时的文本内容大幅精减，再对停用词进行更新，进行一次去停用词处理，可以较好地解决这个问题。这部分内容不是本文的重点，简略）。

2.2 时态文本表示

在对时态文本进行预处理后，需要将已清理的时态文本进行表示。在文本处理时我们已提取论文的发表时间，所以将时间和文本分列处理，只需将文本进行表示。文本的特征表示是指以一定特征项（如词条或描述）来代表文档，在文本挖掘时只需对这些特征项进行处理，从而实现非结构化的文本处理，这是一个非结构化向结构化转化的处理步骤。特征表示的构造过程就是挖掘模型的构造过程，其表示模型常见的有：布尔模型、概率性模型、向量空间模型（VSM，Vector Space Model）等，在本文的实验中我们采用VSM，其核心概念可以描述如下：

1. 特征项：组成文档的字、词、句子等。在VSM中我们将文本视为由一组词条（$T_1, T_2, \cdots T_n$）构成，在本文中 T_i 表示一篇医学病毒论文，T_i 在经过预处理后留下的每个单词或词组就记为特征项 i_i。

2. 特征项的权重：在一个文本中，每个特征项都被赋予一个权重，用以表示特征项在该文本中的重要程度，以便提取出文本的关键词。本文用每个特征项出现的次数作为特征项的权重。

3. 向量空间模型：在舍弃了各个特征项之间的顺序信息后，一个文本就表示成向量，即特征空间的一个点。如一条文本记录 T_i 就表示为：

$VSM(T_i)=(i_{i1}*w_{i1}, i_{i2}*w_{i2}, \cdots, i_{ik}*w_{ik}, \cdots, i_{im}*w_{im})$，其中 i_{i1} 表示该文本记录中的某个特征项（一个单词或一条短语），w_{ik} 为权值，表示特征项 i_{i1} 在文本 T_i 中出现的次数。比如本文的实验中编号为20345这篇论文的VSM表示为（5 * yellow 2 * fever 6 * virus 3 * transmitted 2 * monkey 5 * bites ……）。

经过这样的3步操作后，能将每条记录表示成向量空间。在本文的实验中，在对文本进行处理时把每篇病毒论文表示成这样的向量空间模型。

3 时态文本模型

3.1 时态文本表示模型

文献[9]引入了一般意义下的时态型和时间粒度的定义，时态型 μ、时间粒度 v 是描述时态数据挖掘的重要概念。

定义1 对时态文本对象进行研究时，设 T 表示有限个时态文本对象的集合，记 $T=\{T_1, T_2, \cdots, T_f\}$。设 I 是时态文本对象的所有特征项（单词或短语）的集合，也可以说是时态文本对象的属性集，记为 $I=\{i_1, i_2, \cdots, i_n\}$，特征项项集 I 的所有子集构成的集合记为 2^I，$i_i \in 2^I$。

我们经常是在某一时态下考虑事件的，同样也在某一时态下去考虑文本事件。

定义2 设 v 是一个时态型，用 $(T_i, i_i, v(t))$ 表示时态文本对象 T_i 在时态因子为 $v(t)$ 处出现特征项 i_i 的时态文本事件，若特征项 i_i 在对象 T_i 中出现，则记 $E(T_i, i_i, v(t))=1$，否则记 $E(T_i, i_i, v(t))=0$，称 E 为时态文本事件函数。

例如，可以用 $(T_i, i_i, v(t))$ 来表示诸如TID为20456的一篇发表于2010年的文章：batch of aedes stegomyia aegypti which had fed on monkeys in the early febrile stage ……，E（20456，batch，2010）表示batch这个时态文本特征项在2010年发表的这篇编号为20456的文章中出现。

3.2 基于时态文本的关联规则模型

基于时态文本的关联规则体现了同时间粒度下的文本间的关联，以及多时间粒度下的时态文本间的关联，在时态文本数据库中体现为不同记录中所包含的文本项集之间的关联。

定义3 给定绝对时间段 $[T, T']$，不妨设 v 将时间段 $[T, T']$ 划分为 n 段，即存在 n 个 $t_1, t_2, \cdots, t_n$，使得 $[T, T']=\bigcup_n v(t_i)$，其中 $t_1<t_2<\cdots<t_n$，$v(t_i)\cap v(t_j)=\phi$，$i\neq j$，$i, j=1, 2, \cdots, n$。

把 $(T, 2^I, [T, T']v)$ 称为时态文本对象的特征在时间区间 $[T, T']$ 上关于时态型 v 的文本事件空间，时态文本事件空间由所有的文本事件 $(T_i, i_i, v(t))$ 构成。

这里给出时态文本事件空间中的任意两个事件的逻辑蕴涵关系，设任意两个不同的时态文本事件为 X, Y，有定义如下。

定义4 若 $E(X, Y)=1$，则记 $X\rightarrow Y$，称之为时态文本关联规则。

以下均讨论在时间区间 $[T, T']$、时态因子为 $v(t)$ 下发生的时态文本事件。如果 $v(t)\subset[T, T']$，则认为事件 $(T_i, i_i, v$

$(t))$都不发生，$E(T_i,i_i,v(t))=0$。

定义5 设$k=1,2,\cdots,[T,T']$中的时态因子为$v(t_k)$，有文本$(T_i,\bar{i},v(t_k))$和$(T_i,\bar{\bar{i}},v(t_k))$，且$\bar{i}\cap\bar{\bar{i}}=\phi$，则记时态文本对象$T_i$在同一时态因子处有不同的特征项。定义$E((T_i,\bar{i},v(t_k))\wedge(T_i,\bar{\bar{i}},v(t_k)))=1$表示$(T_i,\bar{i},v(t_k))$发生时$(T_i,\bar{\bar{i}},v(t_k))$也发生，也就是$(T_i,\bar{i},v(t_k))\rightarrow(T_i,\bar{\bar{i}},v(t_k))$成立，称之为时态文本关联规则，简记为$(T_i,\bar{i})\xrightarrow{[T,T']v}(T_i,\bar{\bar{i}})$。

符号$(T_i,\bar{i})$表示在时间段$[T,T']$中存在时态因子$v(t)$使得文本事件$(T_i,\bar{i},v(t))$发生，即时态文本事件可简记为$(T_i,\bar{i})$。记$N^{v}_{[T,T']}(T_i,\bar{i})=\sum_{k=1}^{n}E(T_i,\bar{i},v(t_k))$，表示$(T_i,\bar{i})$在时间段$[T,T']$中的所有时态因子$v(t)$下，时态文本事件$(T_i,\bar{i},v(t))$重复发生的次数(即特征项$\bar{i}$重复出现的次数)。

为了标明时态文本事件$(T_i,\bar{i})$在$[T,T']$中发生的区间，引入“有效时间”的概念如下。

定义6 时态文本事件$(T_i,\bar{i})$在$[T,T']$中第一次发生的时间因子$v(t_{Begin})$和最后一次发生的时间因子$v(t_{End})$所构成的时间区间$[v(t_{Begin}),v(t_{End})]$称为文本事件$(T_i,\bar{i})$的有效时间。项集$i(i\subseteq I)$记为$i^{B,E}_{count}$，$(B\leqslant E)$，其中$B$表示该项集出现的时间，$E$表示该项集结束的时间，$count$表示项集$i$出现的频数。

定义7 由于大型时态文本数据库的数据海量，并且具有实时更新性，在挖掘时态关联规则时我们将数据集D按时间粒度划分为n个阶段部分，时间粒度可以是一年、一个月、一天等[10]，每个阶段部分记为S_i。假设$db^{B,E}$为时态数据库中的一部分，那么把$db^{B,E}$称为时段数据库，$db^{B,E}$可看成由连续的$S_B,\cdots,S_E$组成，且

$$|db^{B,E}|=\sum_{h=B}^{E}|S_h|$$

其中，$db^{B,E}\subseteq D$。

如果项目集X$(X\subseteq D)$中的所有项最早同时出现的时间为B，最晚同时出现的时间为E，那么项目集$X^{B,E}$在时段数据库$db^{B,E}$中称为最大时态项目集(TI)。一个项目集里各子项共同存在的最大时间用MCET(Maximal Common Existing Time)来表示，称为最大共同存在时间。那么用$MCET(X)$表示项X的MCET值。项目集X的MCET值在项目集X的所有子项目中是最短的。

若在数据库$db^{B,E}$的某时段中，有K条记录包含项集$\{X_1,X_2\}$，则K就是项集$\{X_1,X_2\}$在该时段下的支持数。时段数据库$db^{B,E}$中的文本事务数记为$|db^{B,E}|$，文本事务集为T，项集为X，那么$X^{MCET(X)}$在$db^{B,E}$中的支持度为：

$$\sup(X^{MCET(X)})=\frac{|\{T\in db^{MCET(X)}\mid X\subseteq T\}|}{|db^{MCET(X)}|}$$，规则$X\rightarrow Y^{MCET(XY)}$的支持度定义为：

$$\sup(X\rightarrow Y^{MCET(XY)})=\sup((X\cup Y)^{MCET(XY)})$$

该规则的置信度为：

$$con(X\rightarrow Y^{MCET(XY)})=\frac{\sup((X\cup Y)^{MCET(XY)})}{\sup(X^{MCET(X)})}$$

按这样的方法，时态关联规则就可以用如下方法进行定义。

定义8 我们用$(X\rightarrow Y)^{MCET(XY)}$表示事务集$D$中的一条时态关联规则。如果$db^{MCET(XY)}$中$s\%$的事务含有$X\cup Y$，且$X\cup Y=\phi$；$c\%$的事务在含有$X$的同时也含有$Y$。那么支持度表示为：$\sup((X\rightarrow Y)^{MCET(XY)})=s\%$；置信度表示为：$con((X\rightarrow Y)^{MCET(XY)})=c\%$。在每条关联规则的最大存在时间范围内，已知min_sup和min_con为最小支持度阈值和最小置信度阈值，挖掘时态关联规则的问题就是找出所有的强时态关联规则。由此，用如下方法来定义强时态关联规则。

定义9 时态关联规则$(X\rightarrow Y)^{MCET(XY)}$，若$\sup(X\rightarrow Y)^{MCET(XY)}\geqslant$min_sup，且$con(X\rightarrow Y)^{MCET(XY)}\geqslant$min_con，则称$(X\rightarrow Y)^{MCET(XY)}$为强时态关联规则，否则称为弱时态关联规则。

定义10 当最大时态项目集$X^{MCET(X)}$的出现次数大于给定的最小支持数时，$X^{MCET(X)}$是频繁的。

性质1 若最大的时态k项集$X_k^{MCET(X_k)}$在$db^{MCET(X)}$项目集中是频繁的，它的每一个子项集$X_i^{MCET(X_k)}$$(1\leqslant i\leqslant k)$在$db^{MCET(X)}$中也是频繁的。

由此可知，时态关联规则的挖掘划分为两个子问题：1)根据最小支持度找出对象T_i中的所有频繁项集；2)根据频繁项集和最小支持度产生强关联规则。

4 问题描述

关联规则第二个子问题很简单，可以有效地在一个合理的时间内完成[6]，然而第一个子问题仍非常繁琐，这个过程的时间复杂度仍然很高[13]。对于已发现的关联规则也早已提出了有效修剪规则的算法[14]。

近年来，对文本数据挖掘的研究越来越受到关注，我们发现许多算法都不能有效地运用到文本数据挖掘中。目前已提出的算法不能运用到该数据库的原因是：1)该数据库对象为时态文本对象，需要先对时态文本进行预处理；2)没有考虑每个独立文本项各自存在的有效时间；3)每个文本项缺少一个合理的支持度数。

例如我们考虑医学病毒论文数据库，由于医学病毒论文数据库是一个时态文本数据库，每篇论文都有自己的有效存在时间，如果只考虑每个项目在整个事务集中出现的次数则会产生不合理结果。我们用下面的例子来解释挖掘时态文本数据库的关联规则时可能产生的问题。

例1 在医学病毒论文数据库中，A、B、C、D、E、F分别指代病毒论文中的6个单词(见表1)，如：在2008年Fileno(该论文数据库中事务的TID为论文编号：Fileno)为28770的这篇论文中单词A出现1次，单词D出现2次。最小支持度和置信度分别设为min_sup=60%，min_conf=75%。

以Apriori算法为基础的算法都具有向下封闭性，所以假如使用现有的关联规则算法去找频繁项集就不合适[9]。比如：虽然项目集$X^{B,E}$不是频繁项集，但并不意味着$X^{B+1,E}$($X^{B,E}$的时态子项集)也不是频繁项集。

通过对表1进行更仔细的观察，在文本数据的时间上会发现两个问题：1)早期发表的医学病毒论文中出现的单词更有可能成为频繁项集；2)随着时间的推移，用户对之前已挖掘出来的规则会降低兴趣度甚至对其无兴趣。在考虑了每个文本项集出现的时间后，本文进一步研究了时态关联规则的算法来解决上述涉及到的时间问题。

表1中数据库D记录了从2008年到2010年的数据，按时间粒度“年”将数据库D划分为$S_1,S_2,S_3$3个时段部分。如何找出支持度和置信度都大于用户事先给定的最小支持度

和最小置信度的强关联规则，可以分解为以下 3 个子问题：

①根据项目集的支持度产生所有频繁的最大时态项目集（*TIs*）。

②针对所有的频繁 *TIs* 计算其时态子项集（*SIs*）的支持度。

③针对所有满足最小置信度的 *TIs* 和 *SIs*，产生出所有的强时态关联规则。

表 1　医学病毒文本数据库

		2008—2010 年数据库								
		Year	Fileno	VSM						
D	S_1	2008	28770	1 * A			2 * D			$db^{1,3}$
			28771		3 * B	1 * C	4 * D			
			28772		2 * B	2 * C				
	S_2	2009	30001		1 * B	1 * C		2 * E		$db^{2,3}$
			30002				1 * D	2 * E		
			30003	2 * A	2 * B	2 * C				
	S_3	2010	49920		2 * B	1 * C		1 * E	2 * F	$db^{3,3}$
			49921		1 * B				3 * F	
			49922	2 * A			3 * D			

因此，本文接下来将集中讨论如何产生频繁的 *TIs* 和 *SIs*。基于先对时态文本进行预处理，再将时态文本数据库按时间进行划分，最后进行时态文本关联规则的挖掘，我们提出了新的算法 SPFM（Segment-Progressive-Filter-Miner），算法具体过程将在下面阐述。

5　时态关联规则算法

5.1　算法综述

在时态文本关联规则计算方法上主要有两个问题：(1)随着文本时态数据库不断更新，关联规则也需要更新；(2)数据库中每个项集的有效时间应允许不同。基于这样的问题，我们提出了 SPFM。该算法主要包括 3 步：1)数据库不断更新；2)对数据库按不同时间段进行划分；3)对每个时间段的事务集挖掘频繁项集。拆分后的数据库，每个阶段部分有不同的支持度阈值，按不同的支持度阈值进行计算来产生候选项集。得到的候选项集有以下两种类型：1)在扫描 S_i 阶段记录时，获得候选项集 *item*(*i*)，如果 *item*(*i*)在 S_i 之前的时间段中曾作为候选项集出现过，则称 *item*(*i*)是 α 类候选项；相反，如果 *item*(*i*)只在 S_i 阶段中才作为候选项出现，而未在 S_i 之前的任意时间段内出现，则称 *item*(*i*)是 β 类候选项。

5.2　算法具体描述

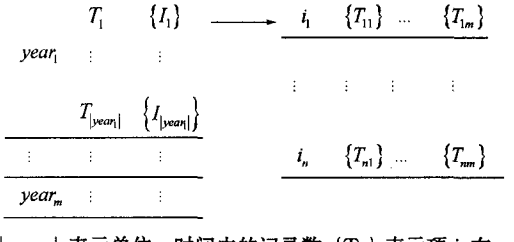
算法　Input：最小支持度 s，最小置信度 minconf，有效时间 ValidT，时态数据库 D

Output：k-频繁项 L_k，时态关联规则集 TAR

1. 算法

①用 OneFreq()函数生成时态数库 D 的 1-频繁项 L_1，并将 1-频繁项转换成垂直数据格式(vertical data formate)，如表 2 所列。

表 2　垂直数据格式转换



其中，$|year_j|$ 表示单位一时间内的记录数，$\{T_{ij}\}$ 表示项 i_i 在 $year_j$ 时间出现的记录集　合，用 $|T_{ij}|$ 表示集合中的记录数，集合可以为空集。

②置 k=2；

while(L_{k-1}！=NULL)　//如果(k−1)-频繁项集已经是空集，则根据性质 1，停止挖掘频繁项集

{

调用频繁项生成算法 Gen_Freq(L_{k-1})；

生成 k-频繁项 L_k，同样采用垂直数据格式保存；

记录最大的频繁项集为 L_{kmax}；

}

③for(k=1;k<=kmax−1;k++)

{

for(j=k+1;j<=kmax;j++)

{

调用关联规则生成算法 Gen_Rule(L_k,L_j)；

将生成的关联规则并入 TAR；

}

}

2. 子算法 Gen_Freq(L_{k-1})

①用 FindInterval(L_{k-1})确定项集 L_{k-1}的有效时间[Begin,End]；

for(j=0;j<m;j++)　//m 是总的时间长度，最小单位为'1'

{

判断 L_{k-1}的$\{T_{ij}\}$集合；

if($\{T_{ij}\}$！=ϕ&&$|T_{ij}|\geq|year_j|*s$)

则令 Begin=时间 j；

for (p=Begin+1;p<m;p++)

{

在时间 j 后寻找有效时间片段的结束时间

End；

if($\{T_{ip}\}\geq|year_j|*s+|year_p|*s-|T_{ij}|$)

则令 $|T_{ij}|+=|T_{ip}|$；如果判断为假，则令 End=p；//即令 $\sum_{j=Begin}^{p}|T_{ij}|\geq\sum_{j=Begin}^{p}|year_j|$

如果判断为假，则确定了 L_k 的一个有效时间片段[Begin,End]，break 跳出循环；

令 j=k，进入下一循环，寻找 L_k 的其余有效时间片段；

}

}

②调用 FindFreq(L_{k-1},Begin,End)在有效时间[Begin,End]内生成 k-频繁项集 L_k

for (j=Begin;j<End;j++)

{

遍历所有的 L_{k-1}；

对于 1 对 k−1-频繁项 L'_{k-1},L''_{k-1}

if($ValidT'_{k-1}\cap ValidT''_{k-1}$！=$\phi$)

取两个时间的交集，并更新 Begin 和 End，继续判断

if(T'_{k-1}，T''_{k-1}项集的重复项=(k−2))

则合并 T'_{k-1}，T''_{k-1}，生成候选 k 项集 C_k；

取更新后的 Begin 和 End，计算这一时间片段对应的支持度数 SC；

$SC=\sum_{s=Begin}^{End}|T_{is}|*s$；

if(support(C_k)≥SC)

则将 C_k 转换成 L_k，并返回保存成垂直数据格式

}

3. 子算法 Gen_Rule(L_k，L_j)

调用 FindInterval()函数确定 L_k，L_j 的有效时间交集；

确定{$L_k \cup L_j$}在时间交集中的支持度，进行置信度、余弦判断；

if(confidence($L_k \xrightarrow{[Begin,End]} L_j$)≥

minconf&& $\dfrac{sup(L_k \cup L_j)}{\sqrt{sup(L_k)*sup(L_j)}}>0.5$)

则关联规则 TAR($L_k \xrightarrow{[Begin,End]} L_j$)为强关联规则，且规则有意义；

同样地，判断($L_j \xrightarrow{[Begin,End]} L_k$)是否为强关联，且有意义；

将判断为真的关联规则 TAR 更新到规则集中。

在本次算法中，调用了 FindInterval(L_{k-1})来确定项集 L_{k-1}的有效时间[*Begin*，*End*]，所以本算法中使用到的有效时间是浮动的，增加了挖掘精度。

以上过程中 SPFM 算法寻找频繁项集有效时间的原理如图 1 所示。

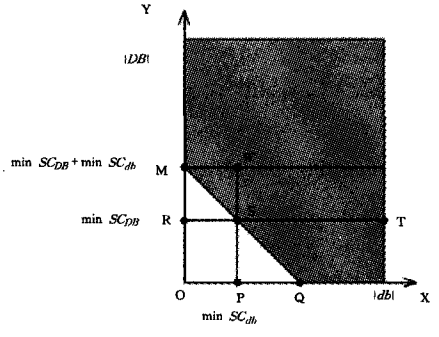


图 1　项集的空间划分

该图将项目集从空间上进行划分，它可以被看作数据库 *DB* 扩大到 *DB*+*db* 时所有可能包含项集 *X* 的记录数，*db* 小于 *DB*。设 *DB* 是时间为 *I* 时的数据库，*DB*+*db* 是时间为 *i* 时的数据库，支持数(记为 *SC*)是这些包含项集 *X* 的记录数。

图 1 简明地描述了 *db* 变动时的情况。该图将数据库 *DB* 和 *db* 中的支持数进行了区分(记 *DB* 中的支持数为 SC_{DB}，*db* 中的支持数为 SC_{db})，图中的每一个点描绘了一个二元组(SC_{DB}，SC_{db})，$SC_{DB}+SC_{db}$ 的值就是在数据库 *DB*+*db* 中的支持数。

图 1 坐标轴中，$X+Y=\min SC_{DB}+\min SC_{db}$，($X\leqslant|db|$，$Y\leqslant|DB|$)。

下面解释图 1 中的几个特殊部分：

因为 $\min SC_{DB+db}=\min SC_{DB}+\min SC_{db}$，所以当数据库 *DB* 扩大到 *DB*+*db* 时，如果支持数没有增加(即 $SC_{db}=0$)，那么为了达到 $\min SC_{DB+db}$，SC_{DB} 就是图 1 中的点 *M*。

图 1 中的直线 MQ 将项目集划分为频繁项集和非频繁项集两部分，图中阴影部分(包括直线 MQ)为所有频繁项集。如果 $\min SC_{DB+db}$，$\min SC_{DB}$，$\min SC_{db}$ 已知，那么可以进一步将图进行划分，图 1 中的每一部分都有各自的意义。直线 MQ 下面的一些部分表示频繁项集仅仅出现在了部分时间段，比如三角形 MSR 表示这些项集在数据库 *DB* 是频繁的，但在 *db* 则是不频繁的，现在在 *DB*+*db* 也是不频繁的。

5.3　算法的时间复杂度

我们设 *m* 为时间数，*N* 为所有的记录数，*n* 为单位时间下的记录数，*L* 为文本数据长度。此算法的时间复杂度为 O($m*N*L+m\wedge 2*L*n\wedge 2+m\wedge 2*n\wedge 2$)。

6　实验测试

为了测试 SPFM 的算法性能，在内存为 2G 的 PC 上用 Visual C++实现了该算法。对象为医学病毒论文数据库中 1970－2010 年间约 50 万条的记录，每条记录的属性包括 fileno(论文标号)、TI(标题)、AB(摘要)、AU(作者)、DP(发表时间)等。由于论文的发表时间与论文审稿、课题研究周期等有关，因此我们认为太小的时间粒度意义不大，所以以“年”作为时间粒度，将数据库划分为 40 个阶段部分。首先设 *D* 为 1970 的文本数据，*d* 为后一年的文本数据，minsup 为 0.5‰，minconf 为 35%，运用前面的算法程序进行频繁项集的挖掘，并确定每个频繁项集的有效时间，依次循环直至 2010 年为止。

首先，对医学病毒论文数据库进行文本处理，包括以下几个步骤：

1)对文本进行预处理：参考第 2.1 节文本预处理部分。

2)提取文本向量：从 AB 和 TI 的合并列 ABandTI 中提取每条记录中单词出现的次数，保留出现 2 次以上(包括 2 次)的单词，作为这一记录的一个文本向量。建 VSM 表，并保存到 VSM 表中，如图 2 红框内部分。

	fileno	DP	vector	Wcount
1	59903	1985	3"efficacy"2"disinfectants"2"antiseptics"2"solutions"	55
2	59904	1985	10"000-da"9"polypeptides"5"monoclonal"4"polypepti...	223
3	59905	1985	10"2-5a"4"poly"2"antibodies"2"possess"2"backbone"...	163
4	59906	1985	10"3a4"5"cells"4"cell"4"antibody"2"monoclonal"2"ale...	267
5	59907	1985	10"accumulation"9"mas"8"ma"4"virus"4"herpes"4"po...	283
6	59908	1985	10"asa"7"virus"5"mice"4"syndrome"4"mortality"3"reye...	186
7	59909	1985	10"aspartokinase"8"subtilis"6"subunits"5"dna"3"codi...	271
8	59910	1985	10"atl"4"ada"3"plasma"3"increased"2"leukemia"2"ad...	157
9	59911	1985	10"avian"10"virus"9"genes"5"replication"4"restricted"...	229
10	59912	1985	10"bovine"9"leukemia"7"virus"5"embryos"4"washed"...	166
11	59913	1985	10"btv"7"infection"7"cattle"5"serotypes"5"goats"5"sh...	279
12	59914	1985	10"capsid"7"tymv"3"virus"3"disruption"3"vrion"3"sta...	237
13	59915	1985	10"cell"5"cells"4"m20"2"myelomonoblastic"2"cytoche...	239
14	59916	1985	10"cell"9"isoenzyme"4"acp"3"1-2"cells"2"acid"2"pho...	269
15	59917	1985	10"cells"3"exocrine"3"endocrine"3"serum"2"tropic"2"...	132
16	59918	1985	10"cells"4"alpha"4"cell"2"responsible"2"vitro"2"subse...	178
17	72013	1985	4"virus"4"hemagglutinin"3"neuraminidase"2"soluble"2...	108
18	72014	1985	4"virus"4"hens"4"shedding"3"peformance"3"liv"3"re...	195
19	72015	1985	4"virus"4"had"4"lambs"2"hairy"2"shaker"	118
20	72016	1985	4"virus"4"hav-2"3"ua"3"nucleotide"3"hav-1"3"herpe...	195

图 2　文本预处理和文本向量 VSM 表

3)进行频繁项集挖掘：从 VSM 表中挖掘出 1-频繁项集，并转换成垂直数据格式，如图 3 所示，一共提取出 924 个 1-频繁项。

	item	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986
1	cells	171	183	231	264	298	1168	1149	1111	1088	1267	1334	1400	1413	1628	1764	1967	1954
2	antigen	36	40	40	53	51	277	228	279	275	271	272	277	308	376	454	426	448
3	infection	71	70	86	84	129	460	479	478	485	537	549	558	568	705	725	886	940
4	hiv	4	6	6	6	5	34	48	46	83	76	76	85	90	112	58	150	135
5	culture	9	22	19	22	17	95	105	70	77	89	85	78	87	117	114	0	0
6	derived	4	0	6	4	11	36	35	29	33	39	44	45	55	56	59	0	0
7	human	20	31	35	47	59	240	244	280	285	321	317	403	416	620	694	19	0
8	antibody	36	39	47	49	54	313	321	331	274	306	334	321	383	452	462	529	565
9	kidney	13	9	12	8	6	42	50	34	32	33	31	34	35	51	39	0	46
10	tissue	12	12	11	13	14	69	58	49	52	44	53	52	63	66	67	83	84
11	hela	4	8	5	6	10	21	30	29	0	0	0	32	27	43	0	0	47
12	groups	7	10	6	11	7	61	63	51	47	63	63	50	76	78	75	0	0
13	cultures	28	39	28	31	26	188	169	96	130	122	119	147	123	157	117	133	154
14	rabbit	8	6	5	7	15	27	31	28	26	0	0	0	0	42	0	0	0
15	guinea	5	6	4	9	5	0	0	0	0	0	0	0	0	0	0	0	0
16	interfer...	35	28	30	24	28	98	112	133	114	141	185	137	147	187	148	163	141
17	produc...	16	22	19	16	23	79	95	67	77	106	104	94	111	123	121	2	0
18	mouse	26	18	28	36	56	145	141	134	158	188	189	182	212	256	236	230	218
19	inducers	4	5	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 3　1-频繁项垂直数据格式

最后使用上文提出的 SPFM 算法挖掘 2-频繁项集、3-频繁项集等直至最大频繁项集，得出时态关联规则 TAR。表 3 列出结果中的几条关联规则。

比如表 3 中的 rous(含铁血黄素)和 sarcoma(恶性毒瘤)，其共同存在时间为：1979 年、1981 年、1983 年，其置信度：conf(rous $\xrightarrow{[1970,1983]}$ sarcoma)＝0.67，conf(sarcoma $\xrightarrow{[1970,1983]}$ rous)＝1，都是强关联规则，而 COS 判断值为 0.8165＞0.5，说明该规则有意义，且这两者在 1979－1983 年是一个共同研究热点，它们之间有可能存在一些密切的联系，在医学上也可以深入研究。同样地，分析 acid(酸)和 deoxyribonucleic(脱氧核糖核酸)，其置信度：conf(acid $\xrightarrow{[1970,1974]}$ deoxyribonucleic)＝0.26，conf(deoxyribonucleic $\xrightarrow{[1970,1974]}$ acid)＝0.84，根据提供的最小置信度判断：conf(deoxyribonucleic $\xrightarrow{[1970,1974]}$ acid)是一个强关联规则，反过来却不是，而对其进行 COS 判断，发现 COS 值＝0.468＜0.5，说明两者的这个关联规则是没有太大意义的，因为 acid(酸)和 deoxyribonucleic(脱氧核糖核酸)本身在含义上有相互包含的关系，这也从一方面证明在进行关联规则挖掘时，引入 COS 判断(余弦判断)有利于筛选出真正有意义的强关联规则。

表 3　时态文本关联规则

Words:A,B	sup(A∪B)	sup(A)	conf (A→B)	sup(B)	conf (B→A)	COS 判断	Begin(A∪B)	End(A∪B)
cells,antigen	2841	104106	0.027	1510	0.18810	0.72	1970	2009
inhibited,virazole	7	7	1.000	7	1.00000	1.00	1973	1973
acid,ribonucleic	38	40	0.950	40	0.95000	0.95	1970	1974
rous,sarcoma	4	6	0.667	4	1.00000	0.82	1970	1983
yaba,poxvirus	4	4	1.000	7	0.57142	0.76	1970	1970
ad2,nd	4	6	0.667	4	1	0.82	1971	1971
avian,myeloblastosis	12	23	0.522	12	1	0.72	1974	1974
salmonella,typhimurium	25	42	0.595	30	0.833333	0.70	1970	1974
equine,venezuelan	15	44	0.341	20	0.75	0.51	1971	1973
coli,scherichia	49	187	0.262	52	0.942308	0.49	1984	1984
sucrose,gradients	5	8	0.625	13	0.384615	0.49	1974	1974
guinea,pig	6	14	0.429	11	0.545455	0.48	1973	1974
equine,encephalomyelitis	5	19	0.263	6	0.833333	0.47	1972	1972
acid,deoxyribonucleic	113	435	0.260	134	0.843284	0.47	1970	1974
sv40,nd	4	19	0.211	4	1	0.46	1971	1971

通过对医学文本数据库的挖掘，可挖掘出上百条时态文本关联规则，从这些规则中能得到近 40 年里学者们对病毒研究的规律以及病毒的发展规律，这些规律是对以往病毒研究的较好总结，也有助于预防病毒产生或更加有效地治疗已产生的病毒。将时态文本挖掘的研究应用于医学病毒领域，是中医学现代化研究的重要组成部分。在文本数据挖掘技术已经日渐成熟的背景下，把时态数据与文本挖掘联合起来，将时态文本数据挖掘应用于医学论文，通过对海量的时态文本数据进行关联分析，总结出一些规律，可以分析治疗规律以及医生的思维模式，辅助医生了解病毒以提高疗效；还能为消灭新病毒和进行实验研究提供目标和思路，减少盲目性，缩短研究周期。通过时态文本挖掘寻找规律和新知识将成为医学研究的热点。

结束语　本文提出了时态文本模型及基于时态文本关联规则的模型，然后提出对医学病毒论文数据库中的时态文本进行预处理：先将时间和文本分为不同的列，将文本表示为向量空间模型，然后确实频繁项集的有效时间，将文本数据转换成垂直数据格式，再通过 SPMF 算法挖掘频繁项集，最后对时态文进行强关联规则的挖掘。该实验是建立在时态文本模型和时态文本的关联规则模型基础之上来运用算法 SPFM 的，充分验证了该算法的有效性。

参 考 文 献

[1] 潘定. 持续时态数据挖掘及其实现机制[M]. 北京：经济科学出版社，2008：36

[2] Agrawal R, Raloutsos C, Swami A. Efficient Similarity Search in Sequence Databases[C]//Proc. of the 4th International Conference on Foudations of Data Organization and Algorithms. Berlin, Germany: Springer-Verlag, 1993: 63-84

[3] 彭丽芳. 基于 SVM 的时态数据挖掘研究[D]. 湘潭：湘潭大学，2006

[4] Gharib T F, Nassar H. An efficient algorithm for incremental mining of temporal association rules[J]. Mohamed Taha, Ajith Abraham Data Knowl. Eng., 2010, 69(8): 800-815

[5] Lee C-H, Lin C-R, Chen M-S. On Ming General Temporal Association Rule in Publication of database[C]//ICDM. 2002: 25

[6] Chen M, Chen S-C, Shyu M-L. Hierarchical Temporal Association Mining for Video Event Detection in Video Databases[C]// IEEE. 2006: 2150-2154

[7] Byon L-N, Han J-H. Fast for Temporal Association Rule in a Large Database. Key Engineering Materials [C]//IEEE. 2005: 279

[8] 韩普，王东波，路高飞. Stemming 和 Lemmatization 对英文文本聚类的研究影响[J]. 信息系统，2012(7)：109-113

[9] 孟志青. 时态数据采掘中的时态型与时间粒度研究[J]. 湘潭大学自然科学学报，2000，22(3)：1

[10] 宁慧. 基于时态约束的关联规则挖掘方法研究[D]. 哈尔滨：哈尔滨工程大学，2007

[11] Tansel A U, Imberman S P. Discovery of Association in Temoral Databases[C]//IEEE. 2007: 25