基于关联规则挖掘的中文文本自动分类

王元珍[1]，钱铁云[1]，冯小年[2]

[1](华中科技大学 计算机学院 数据库与多媒体技术研究所，湖北 武汉 430074)
[2](中国电力财务有限公司 华中分公司，湖北 武汉 430077)
E-mail：qty1970@yahoo.com.cn

摘 要：随着电子出版物和互联网文档的飞速增加，自动文档分类工作正变得日渐重要.提出一种基于关联规则的中文文本自动分类方法.该算法将文档视作事务，关键词视作项，利用改进的关联规则挖掘算法挖掘项和类别间的相关关系.挖掘出的规则形成分类器，可用于类标号未知的文档的区分.实验证明，该算法能较快地获得可理解的规则并且具有较好的召回率和准确率.

关 键 词：基于关联的分类；中文文本分类；关联规则挖掘
中图分类号：TP311 **文献标识码**：A **文 章 编 号**：1000-1220(2005)08-1380-04

Association Rules Based Automatic Chinese Text Categorization

WANG Yuan-zhen[1], QIAN Tie-yun[1], FENG Xiao-nian[2]

[1](*Computer Science Department, Huazhong University of Science and Technology, Wuhan* 430074, *China*)
[2](*China Power Finance Company, Huazhong Branch, Wuhan* 430077, *China*)

Abstract: With the rapid expansion of electronic publication, it becomes more and more important to classify document automatically. This paper introduced a new method that is called association based document classification into Chinese Text Categorization. In our algorithms, each document is viewed as a transaction and each keyword as an item, then an association rule mining algorithm is used to mine the correlation between item and category at the end, unlabeled documents are classified using these found rules. Experiments confirmed that this method gets understandable rules of classifier fast and has a promising recall and precision rate.

Key words: association based classification; Chinese text categorization; association rule mining

1 引 言

随着Internet的飞速发展和电子出版物的飞速增长，传统的信息检索技术已经不能够适应大量文本数据处理的需要.对大量电子文档进行有效的过滤并进行自动分类组织，将有助于文档的检索和分析.

自动文本分类是指计算机将一篇文章自动地分派到一个或多个预定义的类别中去.目前国内外在这方面已经展开了相当多的研究.总体上来说目前的方法主要集中在基于统计和基于机器学习的方法.在英文文本自动分类领域已经提出了多种成熟的分类方法，如最近邻分类、贝叶斯分类、决策树、支持向量机、向量空间模型、回归模型和神经网络等.国内中文文本分类技术的研究主要有：基于机器学习的方法[1]；向量空间模型(VSM)[2,7]；支持向量机(SVM)[3,8]，K-最临近分类(KNN)[4]和Boosting[5]，其它还有基于序列的文本分类方法[6]和基于概念的文档分类算法等.从总体上说，中文文本的自动分类技术研究目前尚处于起步阶段，尚未有达到实用要求的系统.我们在进行科技部电子政务项目的开发过程中，在参考了大量文献资料的基础上，提出了基于关联规则的中文文本分类算法，取得了良好的效果.

自动文本分类的过程通常包括两步：第一步，将一组预先分好类的文档作为训练集，并利用一定的分类挖掘算法对训练集中的对象进行分析以导出分类模式，分类模式常用的表现形式有分类规则、判定树或数学公式.第二步是利用获得的分类模式对类别未知的文档进行分类.可以看出，自动文本分类的本质是利用训练文本找出某一类文本中共有的特征，从而将出现某些相同特征的未知文档归入到相应的类别下.关联规则挖掘算法用于挖掘大型事务数据库中项之间的有趣关系，其中最为著名的算法是apriori算法[9]和fp-tree方法，后来Bing Liu提出的基于关联的分类规则挖掘算法CBA[10]通过改造apriori算法，将关联、分类规则挖掘结合起来，设计出基于关联的分类规则挖掘算法.如果我们将文档视为事务，词作为项，那么基于关联的分类规则挖掘可以用于发现某一类文档中共有的特征词，从而利用这些特征词来区分未知文档.

2 相关概念

关联规则开采问题[14]：假设$I=\{i_1,i_2,\cdots,i_m\}$是m个不同

收稿日期：2004-02-19 基金项目：科技部科技电子政务系统关键技术及应用系统的研究(2001BA110B01)资助，高等学校博士学科专项基金(20030487022)资助 作者简介：王元珍，女，1945年生，教授、博士生导师，主要从事现代数据库研究；钱铁云，女，1970年生，博士研究生，研究方向为数据挖掘和信息检索；冯小年，男，1970年生，硕士，主要从事现代数据库研究.

项 (items)的集合.给定一个事务数据库D,其中每个事务T(transaction)是I中一组项目的集合,即$T\subseteq I$.每一条交易都有一个唯一的标识符TID.如果对于I中的一个子集X,有$X\subseteq T$,我们就说一条交易T包含X.一条关联规则(association rule)就是一个形如$X\Rightarrow Y$的蕴涵式,其中$X\subset I$,$Y\subset I$,而且$X\cap Y=\varnothing$.关联规则的开采问题就是生成所有满足用户指定的最小支持度和最小可信度的关联规则.

关联规则 Association rule:形如X⇒Y的蕴含式,其中X⊆I,Y⊆I,且X∩Y=∅

规则支持度(Support):是指D中包含有X并且具有类标号y的实例个数占全部实例个数的百分比.

规则置信度(Confidence):是指D中包含X的实例中被标记为y的百分比.

频繁规则项(Frequent RuleItem):满足用户指定的最小支持度的规则项集.

分类关联规则CAR:是指后件被限定为类别属性的关联规则.设有任务相关的数据集合D,I是项的集合,Y是类标号的集合,分类关联规则是形如X⇒y的蕴含式,其中X⊆I,y∈Y.

Apriori 性质:频繁项集的所有非空子集都必须也是频繁的.

3　基于关联规则挖掘的中文文本自动分类过程

3.1　文本预处理

3.1.1　文本预处理的过程

在基于关联规则的文本分类中,每篇文档被视作一个事务,而文档中所包含的字或词(在此我们统一称之为关键字)被作为项,经过分词、去停用词后将文档转化为事务数据.分词我们采用中科院计算所提供的汉语词法分析系统ICTCLAS.我们按照词性去除停用词,一般只保留名词和动词.因为根据汉语语言的基本知识,名词和动词是一个汉语句子中的核心部分,它们的简单组合已经足够可以标注出文章的主题.

预处理的基本工作是要实现从文档号⇒事务号以及关键字⇒项号的两个映射关系.在此过程中,由于每个独特的关键字都需要赋予唯一的项号,因此在从文档关键字到项号的映射过程中需要形成一个词典保存所有关键字.每个关键字被读入时,首先在字典中匹配,判定是否已经存在,若有则取出对应的项编号,若没有则在词典中加入该关键字并赋予一个新的项编号.由于在新文档的训练过程和利用规则进行分类的过程都需要进行这种转换,所以词典数据结构的设计显得尤为重要,该数据结构的设计必须有利于快速的查找和插入,此外空间结构也非常重要,因为在预处理阶段和进行分类的阶段该词典需要驻留内存.综合各方面的考虑,最终决定选用prefix-hash-tree树作为词典的数据结构.

3.1.2　文本预处理过程中用到的数据结构

英文文本在转化为事务的过程中,可以方便地采用trie-tree或patricia-tree来存储文本中已经出现过的词及其项编号.由于汉语的与英语在语言学特性方面的显著区别,英语只有26个字母,而仅常用汉字就有6763个(据GB2312),若在中文文本处理中采用上述两种结构都需要非常大的内存空间要求.因此在将中文文本转换为事务的过程中,必须采用不同的数据结构.根据比较研究,我们认为将前缀相同的词保存在同一条路径下形成共享前缀的prefix-hash-tree,这种方法一方面可以节省空间,另一方面也有具有快速的查找和插入性能.

由于汉语的首字数量有限,我们将首字设计成连续空间内的节点作为prefix-hash-tree的首层,这样可以使用直接定位法,达到插入和查找的最高效益.为了达到这样的效果,首先需要将GB-2312的汉字编码映射为连续的整数值,并为这些汉字申请具有6763 * sizeOf (Node)的连续空间(即数组).若pChar[0]、pChar[1]中分别存放该汉字的低位和高位字节码,则通过转换方法:(pChar[0]-176) * 94+ pChar[1]就可以获得其数组下标.这样每个节点的存储位置可以直接按照偏移 * sizeof(Node)得到,首字的查找和插入都可以在O(1)的时间内完成.

必须注意到,由于汉字中可以作为词的首字的字数是非常有限的,但是在树的初始构造阶段我们并不能够确知到底哪些可以成为首字.为了获得查找和插入的高效,在此阶段我们假设所有的常用汉字都可以作为词的首字,即以空间的牺牲来获取时间的效益.

对prefix-hash-tree二层以上的结点,我们不再采用跟首层相同的节点类型,而是采用将其内码hash的方式,我们这样做的原因有二:其一是空间开销太大.仅仅以二字词为例就有9000 * 69.8%=6282个(常用9000汉语词组中69.8%为二字词),这些词中即使只有20%的首字不同,也需要125 * 6763 * sizeof(Node)=845375 * sizeof(Node)的空间,这样的空间开销不是我们希望看到的.其二是经过首字查找,再继续定位的时间开销不高,而且越到下层节点具有相同hash值的节点必然越少,因此hash函数的设计也随着层的增加而将InnerCode mod BucketNum中的BucketNum值逐层递减.

根据以上的分析,我们定义prefix-hash-tree的节点类定义为(首层以外节点):

```
class CPrefixTreeNode //prefix-hash-tree 节点
{
  protected:
    long m_nInnerCode;          //汉字内码
  public:
    bool m_bInit;             //标识儿子指针数组是否分配空间
    long m_nItem;             //项值
    long m_nBuckets;          //节点的儿子指针数组的大小(桶数)
    CPrefixTreeNode   * m_pSibling; //指向兄弟节点的指针
    CPrefixTreeNode   * * m_ppChildren;
                                  // 本节点指向孩子节点的hash 表
    typedef  CPrefixTreeNode   * HASHTABLE;
                                  // 定义与本节点相连的hash 表
  }
```

prefix-hash-tree查找和插入算法:

读入词的查找和插入过程就是一个将该词中的字在prefix-hash-tree中逐层匹配的过程.一旦到达叶子节点,如果读入词结束,那么就取得叶子节点中保存的项编号;否则的

话,就将该叶子节点转化为一个内部节点,并创建它的孩子节点,然后将读入词的余下字插入到那些孩子节点中去.限于篇幅,此处不再详述.

3.1.3 文本预处理过程算法.

调用 prefix-hash-tree 查找和插入算法.

输入:经过分词和去停用词的文档数据库

输出:事务数据库

算法:

```
Did=0
root=Φ
nItem=-1
nCurrentMaxItem=0
For each Training Document D do
{
  while (该文档中还有词)
  {
    GetTerm(d, term);//获取文档中的词
    nItem =GetItemNumberOfTerm(root, term)// root 为
                                        prefix-hash-tree 根
    If (nItem! =-1)//如果该词在先前文档中已经出现过(已经赋过
                                        项编号)
    {
      Insert(nItem,AscendingItemLink)//将所得项号插入到链中
    }
    Else                                //如果该词首次出现
    {
        long newItem = InsertTermIntoTree (root, term,
        nCurrentMaxItem);//将该词转换为一个项号并插入到
        prefix-hash-                                tree 中
        Insert(newItem,AscendingItemLink)  //将新项号插入链中
        nCurrentMaxItem++
    }
  }
  Save(Did, AscendingItemLink,TranDB)//将文档 id 及项号保存
                                        到事务数据库
  Did++;
  }
```

3.2 基于分类的关联规则挖掘算法

利用改进后的 Apriori 算法找出所有满足最小支持度和最小置信度要求的分类关联规则 CAR 集合.

输入:已经转换成为事务数据的文档数据库.

输出:分类关联规则的集合 CAR,每一条分类关联规则形式为:I1, I2, …Im⇒Cj. 其中 Ii 为对应于某个词的项编号,Cj 为类别编号.

算法:类似于[10]中算法,此处不再列出.

3.3 类标号未知的文档进行分类(规则的选择问题)

在获得基于关联的分类规则后,接下来可以利用这些规则对类标号未知的文档进行分类.分类的实质是将文档出现的词与在规则前件出现的词匹配,如果匹配成功,则规则后件的类标号就赋给这个文档.但是由于基于关联的分类将挖掘出全部的规则,可以预见,一篇文档适合的规则肯定有多个,如果这些规则都将这篇文档区分到一个类中,则毫无疑问,该文档属于这个类.但是情况往往不是这么简单,也就是一篇文档可能被分到多个类中.这时我们可以采取两种策略:第一是按照规则链的首次适应法,即规则按照(1)置信度(2)最小支持度(3)规则产生的先后次序形成一个规则集合上的全序关系,类似于[10]中所述,一旦文章中含有的词符合全序关系链中某条规则的前件,就将此规则应用到该文档.但这种方法有一个致命的缺陷,让我们考虑以下例子:规则 R1: I1, I2⇒C1, conf=95%, supp=5%, 规则 R2: I1, I3⇒C2, conf=92%, supp=30%, 如果有一篇文档同时符合规则 R1 和 R2 的前件,则到底应该将该文档分到 C1 类还是 C2 类呢?按照[10]则毫无疑问应该利用第一条规则,因为在全序链中它排在前面,但是从实际出发我们认为分到 C2 类可能更合理些,因为尽管 R2 具有稍小的置信度,但是它的支持度却远远大于 R1,即有更多的文档应该属于 C2 类.文[11]讨论了最高置信度方法存在的缺陷并提出利用加权 χ^2 计算规则前件和后件的相关度,从而选择合适规则,但是 χ^2 计算过于复杂,在进行文档分类特别是在线文档分类时并不适合.文[12]引入支配因子(dominance factor)来调节某个类别规则占总规则数的比例以从适合该测试文档的规则集合中寻找合适规则,这种方法同样存在计算复杂的问题.基于这种考虑我们采取第二种策略,即在全部规则的集合中获得适合此文档的所有规则,然后将这些规则按照后件的不同分成 m 个类别,若每个类别中分别含有 $i_1, i_2, \cdots, i_j, \cdots, i_m$ 条规则,记 $\min(\mathrm{PreOfRule}_k)$ 为一条规则的前件项在文档中出现的最小次数,则分别求出每个类别中类别区分度,即规则的置信度 * 支持度 * $\min(\mathrm{PreOfRule}_k)$ 之和 $=\sum_{k=1}^{k=i_j}\mathrm{conf}(\mathrm{Rule}_k) * \mathrm{sup}(\mathrm{Rule}_k) * \min(\mathrm{PreOfRule}_k)$ 公式①,(j=1,2,…,m),然后比较每个类别的类别区分度和值,取其最大者作为该文档的类别.

算法(方法 2):

```
For each Test Document F do
{
   for each Rule R in RuleList
   {
     for each Item I in Rule
     {
       itemCount=I 出现在文档 F 中的次数;
       minPreOfRule 取 itemCount 最小者;
       if(itemCount =0)
       {
         置前件不匹配标志;
         break;
       }
       if(规则项 I 是规则 R 的尾部)
       {
         置前件匹配标志;
         break;
         }
       }
       if(规则 R 前件已经匹配)
         累加规则 R 后件(类别)对应的类别区分度;//类别区分度
                                        定义见公式①
     }
   将类别区分度最大的类别作为文档 F 的类别;
}
```

4 实验比较

在实验设计上,我们采用的 PFR 人民日报标注语料库,该语料库含有环境、计算机、政治等10个类别2815篇文档,字数达17.7M.实验随机抽取其中的90%作为训练集,余下10%

作为测试集,经过分词和去停用词后由训练集、测试集所形成的事务数据库分别含有723734、41820条元组,我们应用关联规则挖掘算法在训练数据库上寻找分类关联规则,然后将这些规则运用到测试数据库上进行分类,所得实验结果如表1所示(本实验在PentiumIV, RAM512M 机器上):其中方法1是直接利用全序规则链进行区分,方法2则利用支持度 * 置信度 * min(规则前件项在文档中出现次数)进行区分.两种方法的支持度和置信度均为0.01,0.5.经过试验,不同的支持度和置信度得到的效果不同,我们取的是其中较好的一组值.

表1　分类结果表

类别名称	准确率 方法1	召回率 方法1	准确率 方法2	召回率 方法2
交通	37.143%	61.905%	43.590%	80.952%
体育	92.308%	80.000%	93.023%	88.889%
军事	72.727%	64.000%	75.000%	72.000%
医药	100.00%	23.810%	100.00%	33.333%
政治	67.647%	46.000%	83.721%	72.000%
教育	57.576%	86.364%	81.818%	81.898%
环境	100.00%	30.00%	90.909%	50.00%
经济	38.180%	90.625%	54.546%	93.750%
艺术	92.857%	52.000%	76.191%	64.000%
计算机	93.333%	70.000%	100.00%	80.00%

系统的时间开销主要集中在文档预处理的过程中,分类规则生成和测试规则只占了2%的时间.注意到预处理的工作完成一次后就可以保存在事务数据库中,以后在规则生成阶段,为了提高分类精度而采取不同的分类策略都可以使用同一个事务数据集,我们认为这样的代价是可以接受的.

5　结　论

本文首次将基于关联规则的分类算法应用到中文文本分类领域,根据上述实验结果我们可以看出,利用关联规则进行中文文本分类具有较好的效果,特别是利用方法2进行分类比利用方法1进行分类效果要好,在10个类别中7个类别的准确率和召回率都有了明显提高,而在其余的3个类中,有两个类在准确率有所下降的情况下召回率有明显的提高,另一个类则召回率略有下降而准确率有大幅的提高,因此从总体上看,我们提出的方法2进行文档区分是卓有成效的.

致谢:感谢中国科学院计算所无私提供的开放源码分词程序.

References:

[1] Huang Xuan-jing, Wu Li-de. Language independent text categorization [J]. Journal of Chinese Information Process, 2000,14(6):1-7.

[2] Liu Shao-hui, Dong Ming-kai. An approach of multi-hierarchy text classification based on vector space model [J]. Journal of Chinese Information Process,2002,16(3): 8-26.

[3] Li Hui, Shi Zhong-zhi. Improving the performance of the text classifier based on support vector machine using the common sense in the text domain [J]. Journal of Chinese Information Process, 2002,16(2): 7-13.

[4] Liu Bin, Huang Tie-jun. A new statistical based method in automatic text classification [J]. Journal of Chinese Information Process, 2002,16(6):18-24.

[5] Shi Tong-nian, Lu Zhong-liang. Research on the Chinese text categorization of multi-classification and multi-label [J]. Journal of Information, 2003,22(3):306-309.

[6] Xie Chong-feng, Li Xing. A sequence based automatic text classification algorithm [J]. Journal of Software, 2002,13(4): 783-789.

[7] He Hai-jun, Wang Jian-fen, Zhou Qing et al. A Chinese web page classifier based on SVM-decision tree [J]. Computer Engineering, 2003,29(2):47-48.

[8] Zhu Hua-yu, Sun Zheng-xing, Zhang Fu-yan. An automatic Chinese-text classifier based on vector space model [J]. Computer Engineering, 2001,27(2):15-17.

[9] Agrawal R, Srikant R. Fast algorithm for mining association rules in large databases [C]. In: Research Report RJ9839, IBM Almaden Research Center, San Jose, Ca, June 1994:1-32.

[10] Liu Bing. Integrating classification and association rule mining [J]. KDD-98, 1998.

[11] Li Wen-min, Han Jia-wei, Pei Jian. CMAR: Accurate and efficient classification based on multiple class-association rules [C]. ICDM2001:369-376.

[12] Osmar R Zaiane, Maria-Luiza Antonie. Classifying text document by association terms with text categories [C]. The Thirteenth Australssian Database Conference (ADC2002), Melbourne, Australia:215-222.

附中文参考文献:

[1] 黄萱菁,吴立德 等.独立于语种的文本分类方法[J].中文信息学报,2000,14(6):1-7.

[2] 刘少辉,董明楷 等.一种基于向量空间模型的多层次文本分类方法[J].中文信息学报,2002,16(3): 8-26.

[3] 李　辉,史忠植 等.运用文本领域的常识改善基于支撑向量机的文本分类器性能[J].中文信息学报, 2002,16(2): 7-13.

[4] 刘　斌,黄铁军 等.一种新的基于统计的自动文本分类方法[J].中文信息学报, 2002,16(6):18-24.

[5] 施彤年,卢忠良 等.多类多标签汉语文本自动分类的研究[J].情报学报, 2003,22(3):306-309.

[6] 解冲锋,李　星.基于序列的文本自动分类算法[J].软件学报, 2002,13(4):783-789.

[7] 贺海军,王建芬,周 青,曹元大.基于决策支持向量机的中文网页分类器[J].计算机工程, 2003,29(2):47-48.

[8] 朱华宇,孙正兴,张福炎.一个基于向量空间模型的中文文本自动分类系统[J].计算机工程,2001,27(2):15-17.