关联规则及其在数字图书馆中的应用研究

余建坤，曾志勇，张文彬
云南财经大学计算机科学系，昆明，650221

摘　要：关联规则是数据挖掘技术中最重要的方法。本文深入研究了关联规则，对关联规则的 Apriori 算法进行了分析和评述，并基于 Visual Basic 6.0，利用其动态数组功能解决了 Apriori 算法性能瓶颈，探讨了关联规则在数字图书馆中的应用，最后以实验数据展示关联规则在读者借阅数据分析中的应用。
关键词：关联规则；Visual Basic；数字图书馆

Association Rules and Application Study in The Digital Library

Jian-kun Yu　Zhi-yong Zeng　Wen-bin Zhang
Department of Computer Science, Yunnan University of Finance and Economics,Kuming ,650221

Abstract: *The Association Rules is the most important method in technology of the data mining. This text further study The Association Rules, has analyzed and commented to Apriori algorithm of The Association Rules. Have realized Apriori algorithm base on Visual Basic 6.0, probe into Apriori algorithm application among the digital library , show with experimental data of application of Association Rules in borrow in the data analysis in readers finally.*
Keywords: *The Association Rules ;Visual basic; the digital library*

1 引　言

1.1 数据挖掘

近年来，随着数据库技术的发展和 Internet 在全球的普及，人们收集了大量的数据。数据是一种宝贵的资源，为了自动地对数据进行分析，自动地发现和描述数据的趋势，自动地发现数据中的知识，迫切需要找到一种新的技术来处理浩如烟海的数据，这种技术就是数据挖掘（Data Mining, DM）技术，目前，数据挖掘技术已成为数据库研究最活跃，最令人激动人心的领域之一。

数据挖掘，又称知识发现（Knowledge Discovery from Database，KDD），它是从大量不完全的、有躁声的、模糊的、随机的实际数据中提取隐含在其中的、人们不知道的、但又是潜在有用的信息和知识的过程。该过程一般由数据准备阶段、挖掘操作阶段、结果表达和解释阶段组成，数据准备阶段又可进一步

分成数据集成、数据选择和数据预处理；挖掘操作包括确定数据挖掘的目标、选择合适的工具、发掘知识的操作和证实发现的知识；结果表达和解释任务不仅要求把结果表达出来，还要求对信息进行过滤处理，如果结果不能令决策者满意还需要重复以上数据挖掘的过程。数据挖掘可以分为总结规则挖掘、特征规则挖掘、关联规则挖掘、分类规则挖掘和聚类规则挖掘等。

1.2 数字图书馆

数字图书馆实质上是基于网络环境的数字资源信息库和数字资源的有效服务。数字资源包括文本、图形、图像、动画、声音等多种媒体，其特点是信息量大，分类复杂，其来源有馆藏数字资源和非馆藏数字资源，数字资源信息库的建设是数字图书馆的基础工程，对于馆藏数字资源，我们可以采用数据挖掘技术实现数字资源的优化建设，建设特色馆藏数据库，非馆藏数字资源主要应用数据挖掘技术基于国际互连网的资源进行建设，所有的资源通过数据仓库技术进行资源整合，并利用数据挖掘技术实现信息服务质量的提升和服务形式的拓展。

关联规则是数据挖掘技术的重要方法。下面我们将对关联规则的 Apriori 算法进行分析和评述，并介绍 Visual Basic 6.0 实现关联规则算法的编程要点，探讨关联规则在数字图书馆中的应用，最后以实验数据展示关联规则在读者借阅数据分析中的应用。

2 关联规则

关联规则是发现交易数据库中不同商品之间的联系，发现这样的规则可以应用于商品货架设计以及根据购买模式对用户进行分类。Agrawal 等于 1993 年首先提出了挖掘顾客交易数据库中项集间的关联规则问题，以后诸多的研究人员对关联规则的挖掘问题进行了大量的研究，他们的工作主要对原有的算法进行优化，如引入随机采样、并行思想等，以提高算法挖掘规则的效率或者对关联规则的应用进行推广。

2.1 关联规则的概念

设 $I=\{i_1, i_2,\ldots, i_m\}$是二进制文字的集合，其中的元素称为项。记 D 为交易 T 的集合，这里交易 T 是项的集合，并且 $T\subseteq I$ 。对应每一个交易有唯一的标识，如交易号，记作 TID。设 X 是一个 I 中项的集合，如果 $X\subseteq T$，那么称交易 T 包含 X。

一个关联规则是形如 $X\Rightarrow Y$ 的蕴涵式，这里 $X\subset I$, $Y\subset I$，并且 $X\cap Y=\Phi$。规则 $X\Rightarrow Y$ 在交易数据库 D 中的支持度（support）是交易集中包含 X 和 Y 的交易数与所有交易数之比，记为 support($X\Rightarrow Y$)，即：

$$support(X \Rightarrow Y)=|\{T:X \cup Y \in T,\ T \subseteq D\}|/|D|$$

规则 $X \Rightarrow Y$ 在交易集中的可信度（confidence）是指包含 X 和 Y 的交易数与包含 X 的交易数之比，记为 $confidence(X \Rightarrow Y)$，即：

$$confidence(X \Rightarrow Y)=|\{T: X \cup Y \in T,\ T \subseteq D\}|/|\{T:X \in T,\ T \subseteq D\}|$$

给定一个交易集 D，挖掘关联规则问题就是产生支持度和可信度分别大于用户给定的最小支持度(minsupp)和最小可信度(minconf)的关联规则。

2.2 关联规则挖掘的算法

Agrawal 等在 1993 年设计了一个基本算法，这是一个基于两阶段频集思想的方法，将关联规则挖掘算法的设计分解为两个子问题：

1) 找到所有支持度大于最小支持度的项集，这些项集称为频集。

2) 使用第 1 步找到的频集产生期望的规则。

为了生成所有频集，使用了递推的方法。其核心思想如下：

(1) L_1 = {large 1-itemsets};

(2) for (k=2; $L_{k-1} \neq \Phi$; k++) do begin

(3) C_k=apriori-gen(L_{k-1}); //新的候选集

(4) for all transactions $t \in D$ do begin

(5) C_t=subset(C_k,t); //事务 t 中包含的候选集

(6) for all candidates $c \in Ct$ do

(7) c.count++;

(8) end

(9) $L_k=\{c \in C_k \mid c.count>minsup\}$

(10) end

(11) Answer=$\bigcup_{k=1}^{m} L_k$;

首先产生频繁 1-项集 L_1，然后是频繁 2-项集 L_2，直到有某个 r 值使得 Lr 为空，这时算法停止。这里在第 k 次循环中，过程先产生候选 k-项集的集合 C_k，C_k 中的每一个项集是对两个只有一个项不同的属于 L_{k-1} 的频集做一个(k-2)-连接来产生的。C_k 中的项集是用来产生频集的候选集，最后的频集 L_k 必须是 C_k 的一个子集。C_k 中的每个元素需在交易数据库中进行验证来决定其是否加入 L_k，这里的验证过程是算法性能的一个瓶颈。

3 采用 Visual Basic 6.0 实现 Apriori 算法

3.1 基本思想

为减少多次重复读取外部数据，提高计算速度，解决算法性能瓶颈，我们利用 Visual Basic 6.0 的动态数组功能将外部交易数据一次性读入数组 a 中，获

取最小支持度的值 minsup，计算出 1 项候选集及其计数，再求出 1 项频繁集及其计数，在此基础上，求出 2 项候选集及其计数，求出 2 项频繁集及其计数。

采用迭代的方法，在 K 项频繁集的基础上生成 k+1 项候选集及其计数（k>=3）。在生成 K+1 项候选集的过程中，根据一个项集是频集当且仅当它的所有子集都是频集进行修剪，以减少 k+1 项候选集，最后生成 k+1 项频繁集，直到 K 项频繁集为空为止。

3.2 程序实现

```
Dim a( ) As String          /*说明动态数组 a，将外部数据读入*/
Dim c1( ) As String         /*一项侯选集*/
Dim l1( ) As String         /*一项频繁集*/
Dim c2 ( ) As String        /*二项侯选集*/
Dim l2( ) As String         /*二项频繁集*/
Dim countc ( ) As Integer   /*侯选集计数*/
Dim countl ( ) As Integer   /*频繁集计数*/
For i = 1 To UBound(aa)
    flag = 0
    For j = 0 To UBound(c1)
        If aa(i) = c1(j) Then
            countc(j) = countc(j) + 1
            flag = 1
            Exit For
        End If
    Next j
    If flag = 0 Then
        ReDim Preserve c1(m)
        ReDim Preserve countc(m)
        c1(m) = aa(i)
        countc(m) = 1
        m = m + 1
    End If
Next I                      /*求出一项侯选集及其计数*/
m = 0
flag = 0
For i = 0 To UBound(c1)
    If countc(i) >= min_sup Then
        ReDim Preserve l1(m)
        ReDim Preserve countl(m)
```

```
            l1(m) = c1(i)
            countl(m) = countc(i)
            m = m + 1
            flag = 1
        End If
    Next I                                  /*得出 1 项频繁集及其计数*/
    If l1(0) <> "" Then
        m = 0
        For i = 0 To UBound(l1) - 1
            For j = i + 1 To UBound(l1)
                ReDim Preserve c2(m)
                ReDim Preserve countc(m)
                c2(m) = l1(i) & " " & l1(j)
                countc(m) = 0
                For n = 0 To UBound(a)
                    If ainb(c2(m), a(n)) Then
                        countc(m) = countc(m) + 1
                    End If
                Next n
                m = m + 1
            Next j
        Next i
    End If                                  /*得出 2 项候选集及其计数*/
    m = 0
    flag = 0
    For i = 0 To UBound(c2)
        If countc(i) >= min_sup Then
            ReDim Preserve l2(m)
            ReDim Preserve countl(m)
            l2(m) = c2(i)
            countl(m) = countc(i)
            m = m + 1
            flag = 1
        End If
    Next I                                  /*得出 2 项频繁集及其计数*/
    ReDim Preserve l1(UBound(l2))
    For i = 0 To UBound(l2)
        l1(i) = l2(i)
```

```
Next i
k = 3
Do While l1(0) <> ""
    m = 0
    For i = 0 To UBound(l1) - 1
        n1 = Split(l1(i), " ")
        For j = i + 1 To UBound(l1)
            n2 = Split(l1(j), " ")
            flag = 1
            For n = 0 To k - 3
                If n1(n) <> n2(n) Then
                    flag = 0
                End If
            Next n
            If flag = 1 Then
                ReDim Preserve c2(m)
                ReDim Preserve countc(m)
                c2(m) = ""
                For n = 0 To k - 3
                    c2(m) = c2(m) & " " & n1(n)
                Next n
                c2(m) = c2(m) & " " & n1(k - 2) & " " & n2(k - 2)
                c2(m) = LTrim(c2(m))
                countc(m) = 0
                For n = 0 To UBound(a)
                    If ainb(c2(m), a(n)) Then
                        countc(m) = countc(m) + 1
                    End If
                Next n
                m = m + 1
            End If
        Next j
    Next I                                          /*得出 K 项候选集及其计
数 */
    m = 0
    flag = 0
    For i = 0 To UBound(c2)
        If countc(i) >= min_sup Then
```

```
            ReDim Preserve l2(m)
            ReDim Preserve countl(m)
            l2(m) = c2(i)
            countl(m) = countc(i)
            m = m + 1
            flag = 1
        End If
    Next I                                  /*得出 k 项频繁集及其计数*/
```

4 关联规则在数字图书馆中的应用

随着信息技术的发展，数字图书馆保存了大量的数据，迫切需要新的技术对数据进行处理；另一方面，随着读者的信息水平和信息要求的不断提高，向读者提供更主动和个性化的服务也摆在图书馆的面前，在数字图书馆丰富的数据背后，必然存在有用的知识可以帮助图书馆管理人员决策。关联规则反映一个事件和其他事件之间的依赖或关联，它是数据挖掘的本质，在数字图书馆的建设中通过关联规则挖掘可对用户每次借阅的文献进行关联分析，可发现各类文献间的关联规则或比例关系，达到优化信息建设或馆藏布局的目的；通过关联分析可以将内部资源和外部资源进行整合，为读者提供丰富的数据资源；通过对历史数据的关联分析，可以达到合理使用资金，优化图书馆资源配置，为读者提供深层次的信息产品；同时可以帮助采购人员确定采购重点，保障图书馆信息资源体系的科学性和合理性。

5 一个应用实例

在读者借阅事务中，每天都产生大量的数据，数据主要由读者证号、借阅时间、条码、索书号、还书日期，操作者等字段组成，从这些数据中，我们试图采用关联规则方法挖掘读者的借阅模式，分析文献之间的联系，更好地为读者服务和进行馆藏资源的布局，一般而言，读者所借阅的文献，与读者所研究的学科有较大的联系，如计算机科学系的读者，大部分借阅计算机类的书籍，数学类的读者，大部分借阅数学类的书籍，那么，在同类读者借阅的文献之间是否存在一定的关联呢？如有关联，我们就可以向同类读者推荐相关书籍，更好地为读者服务，在图书布局时，也可以把相关文献集中，优化图书布局，方便读者和管理员查找。为此，我们以某高校图书馆的数据为基础数据，提取计算机科学系的读者借阅信息，如表 1 所示：

表 1 计算机科学系读者借阅部分信息

借书证号	借阅时间	还书时间	条码	索书号	操作者
2002060001	20040612	20040912	C4456789	TP311.131-43/1099	Liu01
2002060001	20040712	20041206	C1235632	TP311.138-43/4054	Liu01
2002060020	20050608	20050908	C4567234	TP311.138-43/3774	Ma01
2002060010	20060801		C3456782	TP311.138-43/1031	Zhang02
2002060030	20060403	20060703	C8901234	TP274/7402	wangming
2003060010	20060506		C2341234	TP274/1726	wuy
2000060080	20060708		C5678923	TP312C/7437	liming
2003060020	20050203	20050504	C6789341	TP393.092/9082	zhangxiao
2005060012	20060801		C5643213	TP393.4/6092	yuming

根据数据分析的目标，将数据进行预处理，提取借书证号和索书号中的分类号，形成事务数据，如表 2 所示：

表 2 预处理后的事务数据

借书证号	分类号
2002060001	TP311.138，TP311.131，TP274
2002060035	TP393.092，TP393.4，TP311.138，TP311.138
2002060020	TP393.092，TP393.092
2002060010	TP393.092，TP311.138
2002060030	TP393.092，TP393.092

启动 Apriori 算法程序，如图 1 所示，输入最小支持度，单击“读入数据文件”，将预处理之后的事务数据读入数据集文本框，单击“执行算法”，得出一项侯选集、频繁集，多项侯选集、频繁集。

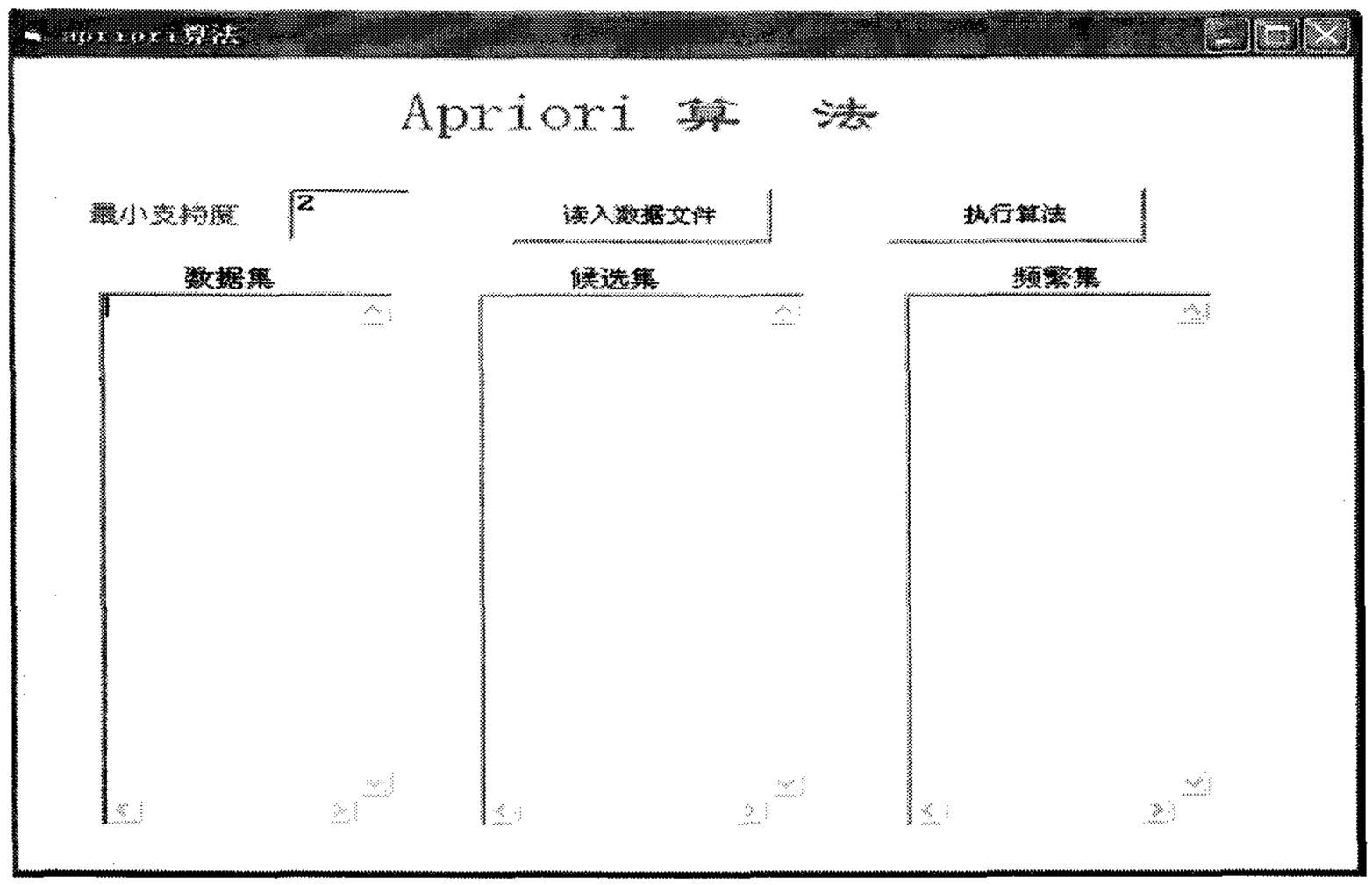

图 1　Apriori 算法程序界面

在所得出的频繁集中，给定置信度，经过计算，我们得到一些有意义的结果，如表 3 所示。

结果表明：在计算机类读者的借阅记录中存在这样的一些规则，借阅了数据库原理书的读者，也同时借阅了 SQL Server 2000 或 Foxpro 类的书籍；借阅了软件工程的读者，也同时借阅了 Visual Basic，Dephi，asp，SQL Server 2000 或 Foxpro 类的书籍；借阅了计算机安全技术的读者，也同时借阅了计算机网络类的书籍；借阅了计算机组成原理书的读者，也同时借阅了汇编语言类的书籍，这样，根据这些信息，作为图书管理人员可以更好地向读者作推介服务，可以优化图书的布局，采购人员可以以此为参考，利用有限的资金进行合理采购。

表 3 关联规则挖掘实验结果

前件	后件	支持度	置信度
TP311.131	TP311.138	0.02	0.04
TP311.131	TP312	0.02	0.04
TP311.5	TP311.138	0.02	0.04
TP311.5	TP312	0.02	0.04
TP311.5	TP393.092	0.02	0.04
TP311.5	TP312BA	0.02	0.04
TP309	TP3393	0.02	0.04
TP309	TP393.08	0.02	0.04
TP302	TP313	0.02	0.04

6 结　语

Apriori 算法是关联规则的经典算法，但是存在多次 I/O 操作，影响计算速度，许多学者进行了研究，本文采用动态数组将数据一次性读入内存，提高了计算速度，但是对于大数据量的处理，这是一个值得进一步研究的问题，数字图书馆是图书馆的发展方向，关联规则的数据挖掘技术将在数字图书馆的建设中取到重要的作用。

参考文献：

[1] Jiawei Han and Micheline Kamber, Data Mining:Concepts and Techniques, Morgan Kaufmann Publishers, Inc.,2001

[2] 彭仪普、熊拥军，关联规则在文献借阅历史数据分析中的应用，情报技术，No.8，2005

[3] 黄　兰，数据挖掘技术在图书馆工作中的应用，图书馆学研究，2005.7

[4] 王共予、李月丽，数据挖掘技术与数字图书馆建设，现代情报，2002.9

作者简介：

余建坤（1962—），男，教授，云南财经大学计算机科学系，研究方向为计算机应用。

曾志勇（1980—），男，博士，云南财经大学计算机科学系，研究方向为计算机应用。

张文彬（1961—），女，副教授，云南财经大学计算机科学系，研究方向为计算机应用。