计算机科学1999Vol. 26№ 12

基于 Group-By 原语接口的分类树挖掘算法的应用

An Application of Classification Tree Method Based on Group-By Primitive Interface Protocol

高 飞 黄敬雄 谢维信
(深圳大学信息工程学院 深圳 518060)

Abstract Recent years, data mining technology has widely been used in many areas, but the widely ignored step during the study of data mining is data extraction problem and the problem of the interface between data mining algorithm and databases. This paper presents a Group-By Primitive Interface Protocol (GPIP) as the interface between data mining algorithm and databaes, and illustrates its using by an example where the mining algorithm is classification tree.

Keywords RDBMS, Datawarehouse, Data mining, Data extraction, Classification tree

一般的传统数据挖掘是，先将与挖掘有关的数据库中的各属性(字段)的值抽取出来，并将其存入平式文件中(即只有单一关系的文件)、然后将这些平式文件装入内存，再把数据挖掘算法施加于该平式文件，以达到挖掘模式的目的。这样做有很多不完善的地方：①世界上的数据每20个月增加一倍[1]，数据库的大小和数量与日俱增，对数据的复制工作往往要占用大量的磁盘空间、时间，并且由于数据的复制而带来一个无法使被挖掘数据保持与原数据库的同步问题。②数据挖掘是一个循环的过程，因此，一旦人们对被考查的数据进行重新定义后，数据的提取工作又要重新进行。③许多文件系统对文件的大小都有限制，一般要小于2GB，人们不得不对大型的数据库进行分割，从而需维护一个文件集。④对抽取出来的数据文件能否施加一个数据挖掘算法也是事先不能确定的问题，因为很多算法往往要求将文件的内容先装入内存，如果抽取出来的文件过大，这几乎是不可能的。实际上，数据挖掘工作中，与数据库的全部内容打交道不但累赘不堪，也是完全没有必要的，因为大多数的数据挖掘算法的实施仅依赖于数据库中的统计信息。本文介绍的 Group-By 原语很好地解决了数据库挖掘算法与数据库间的接口问题，克服了传统挖掘方式的不足。此外，在分类树挖掘算法中的一个关键便是如何从众多的待选属性中挑选出最优属性的问题，因为它直接关系着最终分类模式的好坏。本文用信息增益率(Information Gain Ratio)作为属性的选择标准，它也是 C4.5分类树算法[2]中常用的分类属性选择标准，而计算信息增益率所需的信息则完全可由 SQL 查询给出。

1. Group-By 原语

定义数据挖掘算法与关系数据库的接口的首要问题是确定数据挖掘算法需要什么样的信息。大多数数据库挖掘算法与数据库的交互作用仅仅是发出一系列为求得条件概率的查询，例如贝叶斯分类算法只是利用了与类和属性有关的二元统计信息查询[3]，即为求得条件概率 $P(T|A)$ 的查询，其中 T 为被预测的目标属性，A 为与预测 T 有关的相关输入属性向量，运行该算法所需的时间都集中在对被挖掘数据库的扫描和计算这些统计信息。构成 Group-By 原语接口的 SQL 查询足以提供为计算这些条件概率所需的统计信息。

定义：Group-By 原语接口(GPIP，Group By Primitives Interface Protocol)是指利用 SQL 查询实现数据挖掘算法与数据库管理系统接口的一种模型，在这种接口中，数据库挖掘算法所发出的查询应满足如下形式：

```
SELECT T,A1,…,An,AGG(*)
FROM R
WHERE C
GROUP BY T,A1,…,An
```

这里，T 代表 Target Attribute 或 Class Attribute，即指被预测的类属性，$A_1,\cdots,A_n$ 代表 Attribute，即指与

高 飞 博士生，讲师，主要研究方向：数据挖掘、知识采集、神经网络与 GA 算法、模糊数据库。黄敬雄 博士后，副教授，主要研究方向：神经网络与 GA 算法、小波分析、模糊数据库。谢维信 教授，博士导师，校长，主要研究方向：雷达目标识别、模糊聚类分析，模糊模式识别等。

预测有关的其它属性。AGG 代表 Aggregation，它可以是 count，min，max，sum 之一，R 代表 Relation，即关系或称表，C 代表 Constraints，即对关系施加的约束。

考察有这样一个关于学生信息的数据库，其目的是：通过对测试样本空间的数据挖掘得出类似于"性别＝男，入学成绩＝差的学生中不及格的概率为50%"这样的模式，从而对符合该类学生特点的学生采取提前加强辅导等措施，达到降低不及格率的目的。本例中我们可以发出类似于如下形式的 SQL 查询：

```
>SELECT 及格，入学成绩，性别 COUNT( * )
FROM STUDENT
WHERE 年级＝2，专业＝理科
GROUP BY 及格，及学成绩，性别
```

SQL 查询所得的统计信息如下：

及格	入学成绩	性别	COUNT
yes	好	男	31
yes	好	女	22
yes	差	男	3
yes	差	女	8
no	好	男	3
no	好	女	1
no	差	男	6
no	差	女	2

图1

如图1所示，SQL 里的一个 GROUP BY 查询的 COUNT 列返回的是关于符合所选属性值的任意组合的元组的频繁集（各属性值组合的统计信息），而数据挖掘算法正是利用这些数据来计算特定的概率分布的。在下一节中，我们将详细讨论 GPIP 在分类树算法中用于从预选的分类属性中选取最佳分类属性的应用。下面我们仅对图1中的参数项加以说明。

在图1中，T＝及格，A_1＝入学成绩，A_2＝性别，AGG 为 COUNT，R 为 STUDENT，C＝{年级＝2，专业＝理科}，GROUP 按 T，A_1，A_2的顺序分组。需要说明的是，本例中我们是从整个数据库的一个子集 C＝{年级＝2，专业＝理科}进行数据提取的，如果 C 为空的话，那么数据的提取将是对整个数据库的操作。另外，这里的 R 只涉及到一个关系 STUDENT，如果需要多个关系的话，我们可以创建一个将多个关系联接起来的视图，并将该视图指定给 R 即可。

2 分类树挖掘算法中最佳分类属性的选择

大多数分类树算法都属于自顶向下、回归划分的。算法开始于一个完整的训练样本集，通过对某一属性值的回答，将元组集分为两个子集，接着再对子集进行进一步的划分，直至所有的子集不需要再划分为止，这些不再继续划分的子集被称为叶子，这些叶子被冠以不同的类名，找出相应类的统计特性（类特征），实际上便已初步完成了分类模式的发现工作。建立分类树的目的就是在给定关系中的其余属性后，对类属性进行预测，而对叶子所取的类名是依据该叶子中统计频率最高的类。在构造一个分类树时，一个十分关键的问题就是如何从众多的待选属性中选出最佳的用于分类的属性，因为分类属性的正确选择决定了最终分类模式的优劣。几乎所有的分类准则都是关于类和预选属性的统计信息的函数（见[4]），在分类树挖掘算法中可以采用很多方法作为分类准则的判定准则，比如可采用 χ^2 统计度量，或采用信息增益率（IGR）统计度量。本文以 IGR 作为分类准则的判别依据，进而阐明在分类树挖掘算法中如何利用 GPIP 作为算法与数据库交互的接口。通常，一个 SQL 查询的结果可表示为(m＋1)＊(n＋1)维矩阵结构：

$$\begin{bmatrix} m_{11} & \cdots & m_{1n} & m_{1+} \\ \cdots & \cdots & \cdots & \cdots \\ m_{m1} & \cdots & m_{mn} & m_{m+} \\ m_{+1} & \cdots & m_{+n} & m_{++} \end{bmatrix}$$

图2

说明：m_{ij}（$i<m, j<n$）代表 T 属性取其值集中的第 j 个属性值、属性 A 取其值集中的第 i 个属性值时，被测试的样本空间中满足条件的元组的个数。$m_{1+}\cdots m_{m+}$ 分别为前 m 行的和，$m_{+1}\cdots m_{+n}$ 分别为前 n 列的和，m_{++} 代表被测样本空间中的所有元组的个数。

信息增益率（IGR）的数学定义如下：

$$IGR(A)=IG(A)/I(A) \qquad (1)$$

其中：$IG(A)=I(T)-I(T|A)$

$$I(T)=-\sum_{i=1}^{n}P(T_i)\log_2P(T_i)=-\sum_{j=1}^{n}(m_{+j}/m_{++})\log_2(m_{+j}/m_{++})$$

$$I(T|A)=\sum_{i=1}^{m}P(A_i)\left(-\sum_{j=1}^{n}P(T_j|A_i)\log_2P(T_j|A_i)\right)=\sum_{i=1}^{m}(m_{i+}/m_{++})\left(-\sum_{j=1}^{n}(m_{ij}/m_{++})\log_2(m_{ij}/m_{++})\right)$$

$$I(A)=-\sum_{i=1}^{m}P(A_i)\log_2P(A_i)=-\sum_{i=1}^{i=m}(m_{i+}/m_{++})\log_2(m_{i+}/m_{++})$$

由上述 IGR 的数学定义可以看出，用于计算 IGR 的所有信息全部包含在图2的矩阵中，I(X)的物理意义是度量某一属性 X 的取值在给定元组集下的不确定度，IG(A)＝I(T)－I(T|A)，说明 IG(A)越大，按 A 属性分类的可信度越高，分类越理想。下面让我们回到本文关于学生信息的分类的例子中来。图3描述的是分类树的

建立情况。

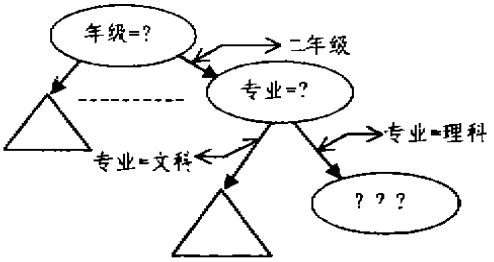

图3

在图3中的???号节点处，我们应该采用哪一可选属性作为分类属性呢？现在有两个可选属性：入学成绩、性别，这个问题的回答实际上就是比较IGR(入学成绩)与IGR(性别)的大小问题，按IGR取值最大的属性进行分类。本例中，对应于两种分类属性的矩阵分别如下：

$$\begin{bmatrix} 34 & 9 & 43 \\ 30 & 3 & 33 \\ 64 & 12 & 76 \end{bmatrix} \qquad \begin{bmatrix} 53 & 4 & 57 \\ 11 & 8 & 19 \\ 64 & 12 & 76 \end{bmatrix}$$

按性别分类　　按入学成绩分类

图4

经IGR公式计算得：IGR(性别)=0.020，IGR(入学成绩)=0.135。由于IGR(入学成绩)>IGR(性别)，故应选择入学成绩属性作为???号处节点的分类属性。按性别分类后，如有必要则继续划分，分类的结果如图5所示。用分类树算法进行数据挖掘的结果是，我们分出了三类学生：Y=3，N=6这一类我们称其为重点辅导的一类，这类学生的不及格率为P=6/9=66.6%；Y=8，N=2这一类我们称其为必要辅导类，这类学生的不及格率约为P=2/10=20%；Y=53，N=4这一类我们称其为无须辅导类，这一类学生的不及格率约为P=4/57=7%。从对数据挖掘的结果分析，可以看出：尽管我们将挖掘的模式分为三类，但其中最为明显的两类是必要辅导及无须辅导这两类，特别是无须辅导类的模式十分明显。因为属于该类的大多数元组都在及格与否的问题上达到了一致，只有7%的例外，也就是说无须辅导这一类的可信度最高。如果一个分类的结果中，所有的叶子的可信度都能达到100%是很困难的(要么都是及格的学生，要么都是不及格的学生)，除非我们知道足够多的属性，并且进行足够多的划分，然而，引入过多的属性和进行太多次的划分会使支持度降低，同时导致算法的复杂化，降低了算法的效率。通常情况下我们最关心的不是支持度和可信度的绝对大小，而是比较关心两者的阈值，模式的发现只要满足给定的阈值即可。当然，在满足一定支持度的前提下，分类的纯度(可信度)越高越好。如果能将上例中的学生只分为两类：即重点辅导和无须辅导，那么分类模式将十分清楚，但这要取决于与问题最为相关的一些属性是否被引入。如果这些属性没有被合理地引入，无论何种算法都不能达到最佳效果，而与GPIP没有任何关系。以上的例子不一定能完全说明实际中的对学生及格与否的估计，因为我们对属性的选择有一定的随机性。但却能从一个侧面反映本文所介绍的GPIP在充当数据库挖掘算法与数据库间的接口时所体现的卓越功能，GPIP的查询结果提供了分类过程中所需的全部信息：即子集划分和属性选择所需的全部信息，当然，它也提供了用于计算分类何时终止的统计度量所需的全部信息。

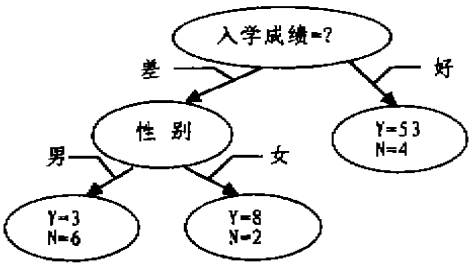

图5

结束语　从上述可以看出：分类树挖掘算法可以用数学的形式表示出来，以便使模型创建的每一步都是基于数据库中的数据的统计信息的函数，这些算法可以利用SQL查询获取所需的统计信息直接实现，如本例用分类树算法创建模型的过程中是基于信息增益率函数IGR的，而计算IGR所需的统计信息则由GPIP完全给出。然而，并不是所有的算法都能在此框架内工作，例如：属于非线性回归的神经网络挖掘算法需要访问每一个数据完整的记录，实际上，在如何利用一个算法时，关键是要明确它需要什么样的信息，GPIP提供了很多算法所需的信息。在数据挖掘里的一个重要问题是大型数据库的适用性问题，很多算法可能在数据库较小的时候尚能得到满意的性能，而一旦数据库大到一定程度就无法使用，对于很大一类算法来说，解决这个问题最直接的方法就是增强DBMS的性能，使其能够提供数据挖掘算法所需的统计信息，以使该算法能够工作在GPIP框架下。另外一种解决的办法就是对查询预先进行计算[4]，这样在数据挖掘的时候算法就可以直接利用这些信息了。当然从算法角度出发，研究适合大型数据库、数据仓库的挖掘算法

(下转第72页)

4.2 可视化的时态约束传递

我们已经知道定性网络存在如下的层次关系[4]：

net(CPA)⊂net(PA)⊂net(IPA)⊂net(IA)=QN

其中：CPA—Convex Point Algebra；PA—Point Algebra；IPA—Interval-Point Algebra；IA—Interval Algebra；QN—Qualitative Network。

net(CPA)、net(PA)、net(IPA)、net(IA)分别表示可用CPA、PA、IPA、IA表示的定性网络。沿着这一层次关系，从net(CPA)到net(IA)，我们可获得更强的表达能力，但同时网络逐渐丧失了易处理性。判断PA网络一致性的复杂度为$O(n^2)$，而判断IPA或IA网络的一致性是NP完全的。

因此，我们可以断定，不存在判断本文提出的一般性网络一致性的多项式算法。针对这种情况，我们首先应用一种基于三角形的定性约束传递算法[6]，它利用时态网络中的道路相容性对时间约束关系进行筛选，其算法的时间复杂性是$O(n^3)$，但是这种方法并不能保证网络的整体相容性。

根据定义3，我们容易看出，如果两个时态对象之间存在着某种定性的时态约束，那么利用它们所允许的区域便可以进行时态对象的裁剪。这实际上是一种弧相容性的检查，它将有助于判断网络一致性问题的解决。

现在假设有n个事件$O_1,O_2,\cdots,O_n$，事件O_i和O_j之间存在定性时态约束R_{ij}。数据集Queue用来记录在约束传递过程中时态视图发生变化的时态对象。在算法开始前，预先将n个事件放在Queue中。可视化时态约束传递算法如下。

```
while Queue is not empty do
  Get next O_i from Queue;
  for j=1,n do
    θ←∅;
    for each r ∈ R_ij do
      if O_j∩δ(O_i,r)=∅
        then R_ij←R_ij−{r};
        else θ←(O_j∩δ(O_i,r))∪θ
      endif
    endfor
    if θ=∅ then exit{signal contradiction};
      elseif θ≠O_j then
      O_j←θ;Queue←Queue∪{O_j};
    endif
  endfor
endwhile
```

需要注意的是，当两个时态对象之间的定性时态约束是多个基本时态关系的析取时，应用上述算法将可能导致在可视时态实体中出现若干个"洞"，即被删除的区域，这在进一步的约束传递中可能会大大增加算法的复杂性。因此，在实际应用中，应该采取一些办法使得两个时态对象之间的定性时态约束只含一个基本时态关系，有关的讨论请参阅文[5,7]。

结论 本文给出了一种集成定性与定量约束的可视时态概念模型。在模型中，事件(时态对象)不再与一个单一的时间区间相对应(一个时间区间只表示事件的一次发生)，而是满足给定约束条件的时间区间的集合。通过在关系数据库中增加特定的时态字段，并且定义一个时态视图，使得每个时态对象可以转化为平面上的一个可视实体，从而导出了可视化的时态约束传递算法。在理论上，该算法并不能保证时态网络的一致性，它只是一种筛选算法。如果我们不仅希望找到一个一致的解，而且希望解是"健壮的"(不会因小小的扰动而失效)，那么本文提出的时态推理的可视化技术将有助于找出这样的解。

参考文献

1 Allen J F. Maintaining knowledge about temporal intervals. Communication of the ACM, 1983, 26(11): 832～843

2 Dechter R, et al. Temporal constraint satisfaction problems. Artificial Intelligence, 1991, 49: 61～95

3 Kautz H, Ladkin P. Integrating metric and qualitative temporal reasoning. In: Proc. AAAI-91. 1991: 241～246

4 Meiri I. Combining qualitative and quantitative constraints in temporal reasoning. In: Proc. AAAI-91, 1991: 260～267

5 Schwalb E, Dechter R. Processing disjunctions in temporal constraint networks. Artificial Intelligence, 1997, 93: 29～61

6 方思行. 时序推理系统TRS的设计与实现. 模式识别与人工智能, 1991, 4(2): 11～18

7 方思行. 一种有效的R-时刻表综合算法. 华南理工大学学报, 1995, 23(9): 43～48

(上接第78页)

也是解决问题的途径之一。总之，数据挖掘算法与数据库间的接口问题仍有很多工作要做。

参考文献

1 Frawley W J, et al. Knowledge Discovery in Databases: An Overview. In: G. Piatetsky-Shapiro & W. J. Frawley, eds. Knowledge Discovery in Databases. Menlo Park, CA: AAAI Press, 1991

2 Quinlan D. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993

3 John G H, Langley P. Estimating continuous distributions in Bayesian classifiers. In: P. Besnard & S. Hands, eds. Eleventh Annual Conf. on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, 1995

4 Cover T M, Thomas J A. Elements of Information Theory, John Wiley & Sons, 1990