应用粗集理论的数据库知识发现

赛 英[1], 赛 煜[2], 张 明[3]

(1. 山东财政学院 计算机信息工程学院, 山东 济南 250014;2. 济南教育学院计算机系, 山东 济南 250001;3. 94070部队, 山东 济南 250117)

摘 要:要从数据量庞大的大型数据库中发现知识,就不得不考虑学习算法的效率。将粗集理论应用到数据挖掘中,实现了从数据库中自动抽取与给定的学习任务相关的属性,能有效地发现简练、贴切的知识,并给出了这一应用的理论基础和实现算法。

关键词:粗集;最小属性集;数据挖掘;KDD

中图分类号:TP3 **文献标识码**:A **文章编号**:1009-3516(2004)03-0068-03

随着计算机技术的高速发展和应用的普及,政府、商业、企业等机构每天都在产生并积累着大量的数据。如何从这些在线数据库中抽取出有用的信息来辅助决策,是一个很有应用价值的课题。目前从数据库中发现知识[1](KDD),或称作数据挖掘,是个十分活跃的研究领域,它是人工智能、机器学习与数据库技术相结合的产物,被数据库以及机器学习研究者誉为20世纪90年代最有前景的研究课题。

现有的许多机器学习算法也适用于KDD,但若应用到数据量庞大的大型数据库中,从效率角度考虑,很多原有算法必须进一步改进。其中一个办法就是简化原有数据库,使其只包含与学习任务密切相关的数据。但经这种化简之后的数据库中仍会包含许多与给定的学习任务无关的或不重要的属性,若能找出相关属性而忽略其它属性,将会大大简化原有关系表,提高学习的效率。Pawlak提出的粗集理论[2-3]为我们提供了一个有效工具,它可以从总体上分析数据库的全部属性,通过将粗集技术集成到学习过程中,我们就可以只考虑与学习任务相关或重要的属性,从而使产生的知识规则特别简练、贴切。

1 粗集理论的分析

1.1 信息系统

四元组 $S=<U,A,V,f>$ 是一个信息系统[3],其中:U 为非空、有限的对象集合,$U=\{x_1,x_2,\cdots,x_n\}$,称作universe。A 为属性的有限集合。A 中的属性又分为两个不相交的子集,即条件属性 C 和决策属性 D,$A=C\cup D$。$V=\cup V_a$, $\forall a\in A$。V_a 是属性 a 的值域。f 为 $U\times A\rightarrow V$ 是一个信息函数,它为每个对象的每个属性赋予一个信息值。即 $\forall a\in A, x_i\in U, f(x_i,a)\in V_a$。

1.2 不可分辨关系

令 $B\subset A$,定义关于属性集 B 的不可分辨关系 $\mathrm{IND}(B)=\{(x_i,x_j)\in U\times U:\forall a\in B, f(x_i,a)=f(x_j,a)\}$。

如果 $(x,y)\in \mathrm{IND}(B)$,则称 x 和 y 关于 B 是不可分辨的。容易证明,$\forall B\subseteq A$,不可分辨关系 $\mathrm{IND}(B)$ 是 U 上的等价关系。

1.3 近似空间

信息系统 $S=<U,A,V,f>$, $B\subset A$, $\mathrm{IND}(B)$ 是关于 B 的不可分辨关系,有序对 $\mathrm{AS}=(U,\mathrm{IND}(B))$ 称作近似空间。等价关系 $\mathrm{IND}(B)$ 将 U 划分为等价类集合,每个等价类中的对象关于 B 是不可分辨的,记作 $U/$

收稿日期:2003-09-15
基金项目:山东省教育厅科技计划项目(J02F06)
作者简介:赛 英(1970-),女,山东荣城人,副教授,博士,主要从事数据挖掘、智能决策等研究.

IND(B)。AS 中任意有限个等价类的并集称为一个可定义集合。令 $X \subseteq U$,定义 X 在 AS 中的下近、上近似集合为:$\underline{\text{IND}}X = \cup \{Y \in U/\text{IND}(B): Y \subseteq X\}$;$\overline{\text{IND}}X = \cup \{Y \in U/\text{IND}(B): Y \cap X = \phi\}$。

由定义,$\forall x_i \in \underline{\text{IND}}X$,$x_i$ 一定属于 X;$\forall x_i \in \overline{\text{IND}}X$,$x_i$ 可能属于 X。即在近似空间 AS 中,X 的下近似是指 X 所包含的最大可定义集合;而 X 的上近似是指包含 X 的最小可定义集合。

1.4 属性集之间的依赖

$C, D \subset A$ 是两个属性集,C 对 D 的依赖程度 $\gamma(C,D)$ 定义为 $\gamma(C,D) = \text{card}(\text{POS}(C,D))/\text{card}(U)$。其中 $\text{POS}(C,D) = \cup CX$, $X \in U/\text{IND}(D)$。$\text{POS}(C,D)$ 称为分类 $U/\text{IND}(D)$ 的正区域。它包含 U 中所有根据属性集 C 的信息可以准确无误地划分到关 IND(D) 的等价类中去的对象。用 card( ) 表示集合中的元素个数。因此 $\gamma(C,D)$ 就表示了能被正确分类的对象在系统中所占的比例,或导师(D)的知识被学习者(C)学习到的百分比。$\gamma(C,D) \in [0,1]$。

1.5 属性的重要度(significance)

$C, D \subset A$,$a \in C$,如下定义 C 中属性 a 关于 D 的重要度:$\text{SGF}(a,C,D) = \gamma(C,D) - \gamma(C - \{a\},D)$。

顾名思义,$\text{SGF}(a,C,D)$ 表示在分类 $U/\text{IND}(D)$ 下,C 中属性 a 的重要程度。

2 粗集理论在 KDD 中的应用

在很多 KDD 应用中,根据决策属性将一组对象划分为各不相交的等价类,我们希望能通过条件属性来决定每一个类。在大多数情况下,通过几个甚至一个属性就决定一个类,而与系统中的其它属性无关。因此,如何找出与决策任务最相关的属性,去除无关和不重要的属性而不丢失任何信息,是一个十分重要的问题。如果我们能找到一个最小的相关属性集,并且它具有与全部属性同样的区分决策属性所划分的类的能力,那么我们就能去除不相关属性,从而简化关系表,并且为每一类产生的规则集将更简练、更有意义。

假设 $S = <U, C \cup D, V, f>$ 是一个信息系统,C 和 D 分别是条件属性集和决策属性集。如果 C 中的某个属性 a 满足 $\text{POS}(C,D) = \text{POS}(C - \{a\},D)$,则 a 关于 D 是多余的;否则 a 是不可缺少的。

若 $B \subset C$ 中的每个属性都是不可缺少的,并且 $\text{POS}(C,D) = \text{POS}(B,D)$,则 B 就是我们要找的最小属性集,称为 reduct。它是一个非冗余的,对可被全部属性集区分出来的所有对象,它也能够同样分辨出来。

需要注意的是 C 的最小属性集一般不唯一。用户可根据不同原则来选择一个“最好”的[4] reduct。如选择具有最少属性个数的 reduct;或根据每个属性的实际含义,为其定义一个费用函数,从而选择具有最小费用的 reduct;再或选择具有最小导出关系的 reduct(即去除冗余属性,相同元组合并后导出的关系最小)。

要找到所有的最小属性集是一个 NP 问题[5],但在大多数应用中,也没有必要找到所有的。下面介绍一个启发式算法,它利用属性重要度和属性间的依赖关系,可在多项式时间内得到一个“最好”的最小属性集。其算法如下(限于篇幅省略部分内容):

```
reduct = ∅;
对 C 中每个属性,根据重要度进行排序;
B = C;
while(POS(reduct,D) ≠ POS(C,D)) do
{
    从 B 中选择具有最大重要度的属性 a;
    if(有多个属性 ai(I = 1…n)具有同样的最大属性重要度)
    {
        计算这些属性 ai 关于 reduct 的重要度 SGF(ai,reduct + {ai},D);
        找其中具有最大重要度的属性 a;
        if(仍有多个属性 ai(I = 1…m)具有同样的最大属性重要度)
        {
            选择其中的 aj,使 aj 与 reduct 中属性导出的关系最小;
            a = aj;
        }
```

```
        }
        reduct = reduct + {a};
        B = B - {a};
        }
    N = |reduct|;
    For i = 1 to N do
    {
        reduct = reduct - {ai};
            if(POS(reduct,D) ! = POS(C,D))
            reduct = reduct + {ai};
```

}该算法在最坏情况下的时间复杂性为 $O(|A| \times |U|2)$。其中$|A|$表示全部属性个数,$|U|$表示信息系统中的对象个数。

得到最小属性集后,我们就可以从原关系表中去除多余的属性,并将相同的元组合并,这样能简化关系表而不丢失任何信息。从简化关系表中将得到更简练的规则集。

3 事例

肺炎与肺结核的一般症状见表 1[6]。根据上述算法,可以得到此信息系统的最小属性集 reduct = {血沉,听诊},由此最小属性集导出的结果见表 2。可以看出,表 2 比表 1 大大简化,这使用户更容易获得各种简练的规则集。比如,我们可获得如下的决策规则集:①血沉 = 正常 ∧ 听诊 = (水泡音 ∨ 干鸣音)→诊断 = 肺炎;②血沉 = 正常 ∧ 听诊 = 正常→诊断 = 肺结核;③血沉 = 快 ∧ 听诊 = 干鸣音→诊断 = 肺结核。

表 1 肺炎与肺结核的一般症状

发烧	咳嗽	X 光所见	血沉	听诊	诊断
高	剧烈	片状	正常	水泡音	肺炎
中度	剧烈	片状	正常	水泡音	肺炎
低	轻微	点状	正常	干鸣音	肺炎
高	中度	片状	正常	水泡音	肺炎
中度	轻微	片状	正常	水泡音	肺炎
无	轻微	索条状	正常	正常	肺结核
高	剧烈	空洞	快	干鸣音	肺结核
低	轻微	索条状	正常	正常	肺结核
无	轻微	点状	快	干鸣音	肺结核
低	中度	片状	快	正常	肺结核

表 2 导出结果表

血沉	听诊	诊断
正常	水泡音	肺炎
正常	干鸣音	肺炎
正常	正常	肺结核
快	干鸣音	肺结核

4 结束语

本文探讨了粗集理论在数据挖掘方面的一个应用,通过研究属性间的依赖程度和属性重要度,找到数据库中的最小属性集,化简数据库,从而有助于得到简练、贴切的规则集。这对于有效地从大型数据库发现知识十分有意义。

当然,粗集理论在 KDD 中的应用远不止此,比如利用粗集理论的近似概念处理数据库中的不精确、不完整信息等。粗集的这些应用领域都具有很大的研究价值和实际意义。

参考文献:

[1] Piatetsky - Shapiro G, Frawley W. Knowledge discovery in Databases[M]. Menlo Park, CA:AAAI/MIT press,1991.

[2] Pawlak Z. Rough Sets[J]. International Journal of Computer and Information Science, 1982, 11(5):341 - 356.

[3] Pawlak Z. Rough Sets:Theoretical Aspects of Reasoning about Data[M]. Dordrecht:Kluwer Academic Publishers, 1991.

[4] Hu X. Knowledge Discovery in Databases:An Attribute - Oriented Rough Set Approach[D]. Canada: University of Regina, 1995.

(下转第 74 页)