文章编号:1001 - 9081(2005)06 - 1350 - 03

一种结合粗糙集和 Cobweb 的聚类器

徐泉清,朱玉文,李 亮,刘万春
(北京理工大学 信息科学技术学院,北京 100081)
(quanqingxu@163.com)

摘 要:提出了一种有效的结合粗糙集和 Cobweb 的聚类算法 CRSC。针对 Cobweb 的不足,引入了粗糙集理论求解属性—值对组的一个最佳归约集,然后结合 Cobweb 算法构建分类树。实验表明,该算法在不降低准确性的条件下,较之传统的聚类算法提高了效率。

关键词:粗糙集;聚类;分类效用;核心;最佳归约集
中图分类号: TP311.131 **文献标识码**:A

Clustering algorithm based on rough set and Cobweb

XU Quan-qing, ZHU Yu-wen, LI liang, LIU Wan-chun
(*the Department of Computer Science and Engineering Beijing Institute of Technology, Beijing* 100081, *China*)

Abstract: An efficient algorithm CRSC(a Clustering Algorithm Based On Rough Set and Cobweb) was proposed. Aiming at the shortage of Cobweb and according to some correlative theories, the theory of rough set was imported to solve a best reduced set of attribute-value pairs, and then it was combined with Cobweb algorithm to construct a hierarchical tree. Our experiment study shows that it greatly advances efficiency without losing accuracy compared with previous methods.

Key words: rough set; clustering; CU; core; best reduced set

在机器学习领域中,聚类分析是一种无指导的机器学习,其中基于模型的方法主要有:统计学方法和神经网络方法。给出一组未标记的对象,概念聚类产生对象的一个分类模式。概念聚类的绝大多数方法采用了统计学途径。

Fisher[1] 提出的 Cobweb:输入对象用分类属性—值对来描述,以一种分类树的形式创建层次聚类;Gennari、Langley 和 Fisher[2] 提出的 ClassIt:用以处理连续性数据的增量聚类;Cheeseman 和 Stutz[3] 提出的 AutoClass:采用 Bayesian 统计分析来估算结果族的数目。我们在探索研究粗糙集理论和概念聚类的基础上,提出了一种结合粗糙集[4] 和 Cobweb 的聚类算法 CRSC。首先利用粗糙集理论进行数据预处理,得到属性的一个最佳规约集并化简,然后根据分类效用,构建分类树。

1 基本概念

粗糙集理论以信息系统(属性—值对)的形式表示数据,对象以属性特征,描述形式与关系数据库相似,且有处理不精确信息、基于示例学习等特点,很适合于数据规约,是一种研究不精确性和不确定性知识的数学工具。

定义 1 不可分辨关系。对于任何一个属性集合 $P \subseteq Q$,不可分辨关系用 IND 表示,定义如下:

$$IND(P) = \{(x,y) \in U \times U : f(x,a) = f(y,a)\ \forall a \in P\}$$

如果 $(x, y) \in IND(P)$,则 x, y 称为相对于 P 是不可分辨的。

定义 2 近似集。对任何一个对象子集 $X \subseteq U$ 和属性子集 $P \subseteq Q$,P 的下近似和上近似定义如下:

$$\underline{apr_P}(X) = \cup \{Y \in U/IND(P) : Y \subseteq X\},$$

$$\overline{apr_P} = \cup \{Y \in U/IND(P) : Y \cap X \neq \varnothing\}$$

定义 3 属性依赖度。在数据规约中,利用两个属性集合 $P, R \subseteq Q$ 之间的相互依赖程度,可以确定一个属性 a 的重要性。属性集 P 对 R 的依赖程度用 $\gamma_R(P)$ 表示。定义如下:

$$\gamma_R(P) = \frac{card(POS_R(P))}{card(U)},$$

$$POS_R(P) = \bigcup_{X \in U/IND(P)} \underline{apr_R}(X)$$

定义 4 属性重要性。不同属性对于决定条件属性和决策属性之间的依赖关系起着不同的作用。属性 a 加入 R,对于分类 $U/IND(P)$ 的重要程度定义为:

$$SGF(a,R,D) = \gamma_R(P) - \gamma_{R-\{a\}}(P)$$

定义 5 属性规约。属性集 $B \subset C$ 是 S 的一个规约,当且仅当 $POS_B(D) = POS_C(D)$,且 B 中的每个属性对于 D 都是不可缺少的。属性规约记为 $RED(B, D)$。

定义 6 核心。相对于属性集 D,C 中所有必要关系组成的集合称为 C 的核心,记为 $CORE(C, D)$。

定义 7 分类效用(Category Utility, CU) 定义如下:

$$\frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right]}{n}$$

这里 n 是在树的某个层次上形成一个划分($C_1, C_2, \cdots, C_n$) 的节点、概念或"种类" 的数目。

2 结合粗糙集和 Cobweb 的聚类算法

收稿日期:2004 - 11 - 12;修订日期:2005 - 02 - 25

作者简介:徐泉清(1980 -),男,安徽东至人,硕士研究生,主要研究方向:数据挖掘、知识发现; 朱玉文(1950 -),女,四川成都人,副教授,主要研究方向:人工智能; 李亮(1977 -),男,湖南祁阳人,硕士研究生,主要研究方向:数据仓库、数据挖掘; 刘万春(1949 -),男,河北滦县人,教授,主要研究方向:模式识别.

2.1 算法设计思想

数据预处理技术可以改进数据的质量,从而有助于提高其后的挖掘过程的精度、性能和效率。由于决策质量的高低必然依赖于数据质量的好坏,因此数据预处理是知识发现过程的重要步骤。检测数据异常、尽早地调整数据,并归约待分析的数据,必然在决策过程得到高回报。

基于粗糙集理论的数据规约,其基本原理是通过求属性重要性并排序,在泛化关系中找出与原始数据具有同样决策或分辨能力的相关属性的最小集合,实现信息约简,以便产生更简洁、更有意义的知识规则。

Cobweb 聚类分析是以一种分类树的形式创建层次聚类。它有如下局限性:

1) 假设在每个属性上的概率分布是彼此独立的,然而假设并不总是成立的;

2) 聚类的概率分布表示使得更新和存储聚类相当昂贵;

3) 分类树对于偏斜的输入数据不是高度平衡的。

基于 Cobweb 的局限性,我们引入了粗糙集理论,对脏的、不完整的和不一致的数据进行数据清洗,提高数据的质量。

2.2 算法依据的主要原理

我们引入粗糙集理论对原始数据进行预处理,不应以降低准确性为代价,换取所谓的高效率。在保证准确性的基础上,我们提出的算法主要依据如下两个定理,提高了算法的效率。

定理 1　若 B 是 C 的一个归约,则有如下性质:

1) 规则 $B\Rightarrow D$ 和 $C\Rightarrow D$ 具有同等强度;

2) 若 $B'\subset B$,则规则 $B'\Rightarrow D$ 严格弱于 $C\Rightarrow D$。

定理2　相对于属性集 D,属性集 C 的核心为属于属性集合 C 的所有规约的交集的属性的集合,即 $CORE(C, D) = \cap RED(C, D)$。

2.3 算法设计和实现

我们提出结合粗糙集和 Cobweb 的聚类算法分为两个阶段。首先在数据预处理阶段,我们引入了粗糙集理论,计算信息系统的一个最佳规约集;然后在执行阶段,构建分类树。

2.3.1 数据预处理

计算属性—值对组的一个最佳规约集

输入:信息系统 S,条件属性集 C 和决策属性集 D,$AR = C$;
输出:一个最佳规约集 $BestRedu$
ComputeBestRedu(S, C, D) {
　初始化:计算核心集 $CORE(C, D)$,并将 $CORE$ 赋给 $BestRedu$,取 AR 中除 $CORE$ 之外其他属性集赋给 AR';
　1) 计算每一个属性 $a \in AR'$ 的 ,并从大到小进行排序;
　2) While ($\gamma(BestRedu, D) \neq \gamma(AR, D)$) {
　　取 $SGF(a_i, BestRedu, D)$ $a_i \in AR'$,$(I = 1,2,\cdots,m)$,并将 a_i 加入到 $BestRedu$,$AR' = AR - \{a_i\}$ $(i = 1, 2,\cdots, m)$;
　　计算 $\gamma(BestRedu, D)$; }
　3) 对于每一个 $a_i \in BestRedu$,如果 $a_i \notin CORE$,则从 $BestRedu$ 除去 a_i,并计算 $\gamma(BestRedu)$; 如果 $\gamma(BestRedu, D) \neq \gamma(AR, D)$,则将 a_i 加入 $BestRedu$;
　4) return $BestRedu$; }

2.3.2 构建分类树

一个未分类的实例 I,将其加入当前节点 N 算法的伪代码:

输入:当前节点 N,未分类的实例 I
输出:概念层次
变量:C, P, Q 和 R 为层次树中的节点,W, X, Y 和 Z 为聚类权值
Cobweb(N, I) {
　If(N 为根节点) 创建新的根节点,并加入节点 N
　Else { 加入节点 N;
　　对于节点 N 的每个子节点 C,计算 C 中 I 的权值;
　　设 P 为最高权值的节点,Q 为第二高权值的节点,X 为 I 产生新节点 R 时的权值,Y 为合并 P 和 Q 为一个节点时的权值,Z 为被分离节点 P 的权值
　　if W 为最大值,then 将 I 放入分类 P 中
　　else if X 为最大值 then 将 I 放入新分类 R 中
　　else if Y 为最大值 then 将 P 和 Q 合并为 O,并将 I 放入 O 中
　　else if Z 为最大值 then 将 N 的子节点 P 移去,并将 P 的子节点提升为 N 的子节点,将 I 放入 N 中 }
}

结合粗糙集和 COBWEB 的聚类算法(CRSC):

输入:信息系统 S,条件属性集 C 和决策属性集 D;
输出:分类树
CRSC(S, C, D) {
　Best_Redu = *ComputeBestRedu*(S, C, D);
　化简最佳规约集 *Best_Redu*;
　对于每一个未分类的实例 $I' \in Best_Redu$ {
　　沿着一条适当的路径向下,根据分类效用,寻找可以分类 I' 的最好节点 N
　　Cobweb(N, I') }
}

2.4 算法实例

我们采用 UCI[7] 机器学习库中的数据集 golf,实值属性离散化采用 MLC ++ 中的熵方法,并采用缺省参数。

表 1　Golf 属性—值对组

U	outlook	temp	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rain	mild	high	false	yes
5	rain	cool	normal	false	yes
6	rain	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rain	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rain	mild	high	true	no

表 2　化简后的 Golf 属性—值对组的最佳规约集

U	outlook	temp	windy	play
1	sunny	hot	false	no
2	sunny	hot	true	no
3, 13	overcast	hot	false	yes
4, 10	rain	mild	false	yes
5	rain	cool	false	yes
6	rain	cool	true	no
7	overcast	cool	true	yes
8	sunny	mild	false	no
9	sunny	cool	false	yes
11	sunny	mild	true	yes
12	overcast	mild	true	yes
14	rain	mild	true	no

在表 1 中,outlook, temperature, humidity, windy 为条件属性,play 为决策属性。首先计算出 CORE,初始化 BestRedu = CORE, AR' = AR - CORE,利用上面求最佳规约集算法得:

BestRedu = {outlook, temperature ,windy},规约后的属性—值对组的最佳规约集,并简化为表 2。

执行阶段,经过零次合并(merging)和一次分裂(splitting),构建如图 1 的分类树(共 18 个聚类节点数)。

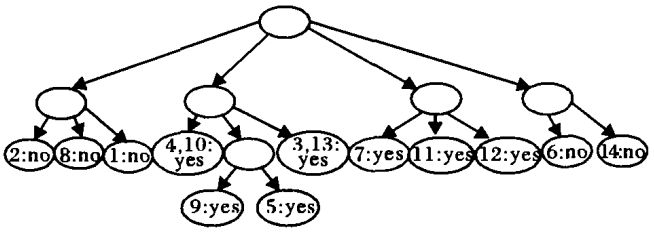


图 1 利用 CRSC 算法创建的分类树

然而,利用 Cobweb 聚类,经过一次合并和一次分裂,构建如图 2 的分类树(共 22 个聚类节点数)。

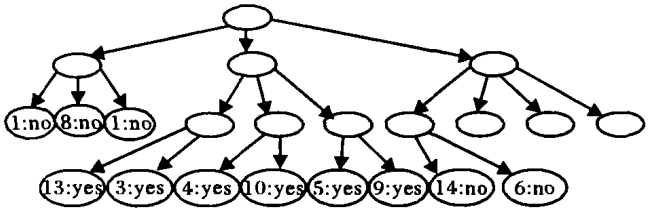


图 2 利用 Cobweb 算法创建的分类树

3 实验结果

我们采用 UCI[7] 机器学习库中的 10 个数据集,技术参数为:合并数、分裂数、聚类节点数和运行时间。前两个参数反映构建分类树时的复杂程度,聚类节点数则反映构建完之后分类树的复杂程度。实验是在运行 Windows 2000,具有 384M 内存,P4 1.6G CPU 的计算机上进行的。

表 3 ZCRSC 算法与 Cobweb 的比较

数据集	Cobweb				CRSC			
	合并数	分裂数	聚类节点数	运行时间	合并数	分裂数	聚类节点数	运行时间
Breast	195	159	692	6940	104	78	457	1753
Breast-cancer	91	67	413	1833	78	55	414	1432
Iris-disc	12	14	22	541	0	0	15	240
Monk1	39	29	196	771	9	5	53	331
Monk3	33	25	190	691	23	14	128	601
parity5 +5	285	216	1600	4707	5	2	44	301
Tic-tac-toe	303	197	1377	4607	329	231	1469	4336
vote	101	80	393	3034	42	26	229	761
Vote-irvine	154	124	531	3274	28	18	192	791
zoo	12	16	65	741	3	1	25	210

在表 3 中,第一列给出了 10 个来自 UCI 的数据集的名字,其后分别 Cobweb 和 CRSC 算法在计算上述 10 个数据集时,合并数、分裂数、聚类节点数和运行时间四个参数的值,其中运行时间的单位为:毫秒(ms)。图 1 至 4 分别为 Cobweb 和 CRSC 在四个参数上的曲线图。

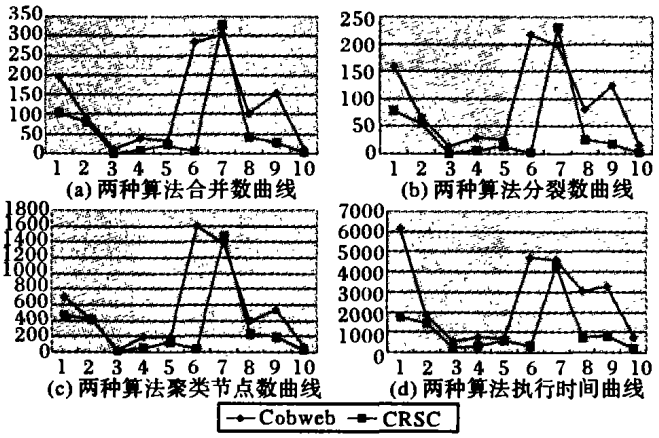


图 3 Cobweb 和 CRSC 在四个参数上的曲线

在处理这 10 个数据集时,我们提出的 CRSC 算法,除了数据集 Tic-tac-toe,其余 9 个数据集在四个参数的指标上都明显好于 Cobweb 算法。我们采用 CRSC 处理 Tic-tac-toe,最佳规约集中的条件属性数目为 8,而原始数据集中的条件属性数目为 9,化简后的最佳规约集与原始数据集中的属性—值对组数目一样,所以 CRSC 与 Cobweb 算法的效果差不多。利用 CRSC 处理数据集 parity5 + 5 后,条件属性数目为 5,属性—值对组数目则变为 32,而原始数据集的条件属性数目和属性—值对组数目分别为 10 和 1024,因此在处理数据集 parity5 +5 时 CRSC 比 Cobweb 效率要好得多。可见,CRSC 算法要比传统的 Cobweb 算法效率要高的多。

4 结语

实践表明,CRSC 算法可以减少数据集中冗余的属性—值组数目,提高了算法的效率。

基于统计学的聚类分析的深入研究和应用还有许多问题亟待解决,例如如何将概念聚类方法更好地应用到数据挖掘中需要进一步的研究;在实际生活中,涉及更多的是模糊概念和模糊知识,如何把它与模糊集理论相结合;以及如何处理随机产生的信息系统等等。

参考文献:

[1] FISHER DH. Knowledge Acquisition via Incremental Conceptual Clustering[J]. Machine Learning, 1987: 139 - 172.

[2] GENNARI JH, LANGLEY P, FISHER DH. Models of Incremental Concept Formation[J]. Artificial Intelligence, 1989: 11 - 61.

[3] CHEESEMAN P, STUTZ J. Baysian Classification (AutoClass): Theory and Result. In advances in Knowledge Discovery and Data Mining[M]. AAAI Press/MIT Press, 1996. 61 - 83.

[4] PAWLAK Z. Rough Sets. International Journal of Computer and Information Sciences[J], 1982, 11(5): 341 - 356.

[5] BODJANOVA S. Approximation of fuzzy concepts in decision making[J]. Fuzzy Sets and Systems 85, 1997: 23 - 29.

[6] NGUYEN HT, WLKER EA. A History and introduction to algebra of conditional events and probability logic[J]. IEEE Transaction Systems, Man, and Cybernetics, 1994, 24: 1671 - 1675.

[7] MERZ C, MURPHY J. UCI repository of machine learning database [EB/OL]. http://www.cs.uci.edu/~mlearn/MLRepository.html, 2000 - 10 - 25/2002 - 02 - 27.