

一种新的启发式粗集决策表属性约简算法

沈 玮, 赵佳宝

(南京大学 工程管理学院 控制理论与系统工程系, 江苏 南京 210093)

摘要:粗集理论通过对原始决策表的约简从而获取规则知识,其核心部分是属性约简。经过约简后的数据更有价值,更能准确地获取知识。文中提出了一种新的启发式属性约简算法,并给出了算法的详细步骤和具体的实验示例。该算法通过不一致计数和互信息增量的计算来衡量属性的重要性,避免了对属性之间随机组合情况的搜索,可以提高求解速度。实验结果表明,相比较于动态约简算法和标准遗传算法,所提出的算法获得的约简属性集更加简洁和高效。

关键词:粗集;属性约简;启发式

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2010)10-0016-05

A New Heuristic Reduction Algorithm of Rough Sets Decision - Making Table

SHEN Wei, ZHAO Jia-bao

(Department of Control & System Engineering, School of Management and
Engineering, Nanjing University, Nanjing 210093, China)

Abstract: Rough set theory acquires rules knowledge through the reduction of the original decision table, and its core part is reduction of attributes. Data after reduction is more valuable and can obtain knowledge more accurately. Presents a new heuristic algorithm, and proposes the detailed steps of the algorithm. And also an example is given to illustrate the algorithm. The algorithm avoids the search for random composition among attributes via using the inconsistency count and the gain of mutual information criteria to value the significance of an attribute, and increases computing speed. From numerical experiments and comparisons, the algorithm provides more precise and simple reduction of attributes than the dynamic reduction algorithm or the standard genetic algorithm does.

Key words: rough sets; attribute reduction; heuristic

0 引言

波兰数学家 Pawlak Z 提出的 Rough Set 是一种新的处理不精确、不完全与不相容知识的数学方法^[1,2]。其目前正被广泛应用于人工智能、模式识别与智能信息处理等领域,并取得了一定的成果^[3]。属性约简是粗糙集理论及应用研究的重要内容之一^[4],也是知识获取的关键步骤。

基于粗糙集的属性约简算法备受研究者的关注并已经取得了一定的成果^[5-7]。最简单有效的方法是对包含强相关属性的离散属性数据集进行属性核的计算,得到的约简包含一个核和另外一些弱相关的属性,从而最后得到满意的属性约简结果。

从所有的属性子集里寻找最优的属性约简集是十分困难的任务,实际上, Wong S K 和 Ziarko W 已证明了寻找决策表的最小属性约简是 NP-hard 问题^[8],而属性的组合爆炸是导致 NP-hard 的主要原因^[9]。为了得到决策信息系统中合适的属性约简集合,文中提出了一个新的启发式属性约简算法。

1 粗糙集相关概念简介

粗糙集理论的要点是将分类和知识联系在一起,并用等价类关系形式化表示分类。可以理解为:知识是使用等价类 R 对离散空间 U 的划分,记为 $U/R = \{X_1, X_2, \dots, X_n\}$,称 X_i 为 U/R 的等价类。

定义 1(决策信息系统) 一个决策表信息系统 S 可以表示为 $S = \langle U, R, V, f \rangle$ 。这里, U 是非空有限对象的集合,也称为论域, $R = C \cup D$ 是非空有限的属性集合,子集分别是条件属性集和决策属性集, $V = \bigcup V_r$ 是属性值的集合, V_r 表示属性 $r \in R$ 的属性

收稿日期:2010-01-30;修回日期:2010-04-17

基金项目:国家自然科学基金(70971063)

作者简介:沈 玮(1986-),女,安徽霍山人,硕士研究生,研究方向为数据挖掘、人工智能;赵佳宝,博士,副教授,硕士生导师,研究方向为数据挖掘、管理控制一体化。

值范围,即属性 r 的值域, $f: U \times X \rightarrow V$ 是一个信息函数,定义对象的属性值。

定义 2(不可分辨关系) 给定决策表信息系统 $S = \langle U, R, V, f \rangle$, 对于每个属性子集 $B \subseteq R$, 定义一个不可分辨关系 $\text{IND}(B)$, 即

$$\text{IND}(B) = \{(x, y) \in U \times U \mid \forall b \in B (V_b(x) = V_b(y))\}, \text{显然不可分辨关系是一种等价关系。}$$

定义 3(上下近似集) 给定信息系统 $S = \langle U, R, V, f \rangle$, 对于每一个子集 $X \subseteq U$ 和不可分辨关系 R , X 的上、下近似集分别可以由 R 的基本集定义如下:

$$R^*(X) = \{x \in U \mid [x]_R \cap X \neq \emptyset\};$$

$$R_*(X) = \{x \in U \mid [x]_R \subseteq X\}.$$

定义 4(相对正域) 设 U 为一个论域, P, Q 为定义在 U 上的 2 个等价关系簇, Q 的 P 正域记为 $\text{POS}_P(Q)$, 定义为 $\text{POS}_P(Q) = \bigcup_{X \in U/Q} P_*(X)$ 。

2 一个启发式属性约简算法

2.1 算法思路

决策信息系统中属性约简的目标就是要从条件属性集合中发现部分必要的条件属性,使得根据这部分条件属性形成的相对于决策属性的分类和所有条件属性所形成的相对于决策属性的分类一致^[10]。

从一个包含很多属性的数据库里得到约简属性子集,一般是一个一个地挑选出最好的属性直到找到一个合适的约简集^[11]。决定一个属性的好坏有很多的评价准则。文中使用不一致计数和互信息增值的计算作为属性约简的评价标准。可以通过衡量从决策信息表中去除一个属性所带来的影响的大小来评价这个属性的重要性。系数 $\gamma(C, D)$ 表示条件属性集 C 和决策属性 D 之间的依赖程度,或是属性集 C 相对于决策属性 D 分类的准确性大小。所以属性 $\{a\}$ 的重要性可以被定义为:

$$\sigma_{(C,D)}(a) = \frac{(\gamma(C, D) - \gamma(C - \{a\}, D))}{\gamma(C, D)} = 1 - \frac{\gamma(C - \{a\}, D)}{\gamma(C, D)}$$

其实系数 $\sigma_{(C,D)}(a)$ 可以被理解成当属性 $\{a\}$ 被去除后所产生的一个分类的误差。当然,这个重要性系数可以被扩展到属性的集合,如下:

$$\sigma_{(C,D)}(B) = \frac{(\gamma(C, D) - \gamma(C - B, D))}{\gamma(C, D)} = 1 - \frac{\gamma(C - B, D)}{\gamma(C, D)}$$

另一个系数 $\epsilon_{(C,D)}(B)$ 称作近似约简的错误率,其表示对于决定决策属性 D 导出的分类,属性集合 B 有多近似条件属性集合 C , 定义如下:

$$\epsilon_{(C,D)}(B) = \frac{\gamma(C, D) - \gamma(B, D)}{\gamma(C, D)} = 1 - \frac{\gamma(B, D)}{\gamma(C, D)}$$

信息熵理论提供了一个评价概率分布变化量的不同方法,其认为论域 U 上的任一等价关系簇是定义在 U 上的子集组成的 σ 代数上的一个随机变量。熵被定义为:

$$\inf o(U) = I(P) = - \sum_{i=1}^k p_i \log p_i$$

这里 U 为论域。如果基于决策属性的值的不同, U 被分割为一些互不相容的子集 $\{Y_1, Y_2, \dots, Y_k\}$, 那么要识别 U 中某个对象的类别所需要的信息就是 $\{Y_1, Y_2, \dots, Y_k\}$ 的概率分布 p :

$$p = (\frac{|Y_1|}{|U|}, \frac{|Y_2|}{|U|}, \dots, \frac{|Y_k|}{|U|}), p_i = \frac{|Y_i|}{|U|}$$

其中 p_i 是这个随机变量所有可能取到的不同的值。 $| \cdot |$ 是集合的基数,这里假设是离散的随机变量,来避免涉及到连续分布系统的熵的专门计算。

互信息 MI 是一个用来测量两个随机变量之间的相关程度的量。两个随机变量 S, R 之间的 MI 通过这两个随机变量的联合分布函数 $p(S, R)$ 来定义。当已经明确这个分布函数时, MI 就可以通过如下公式计算:

$$MI(S, R) = \sum_{s \in S, r \in R} p(s, r) \log \frac{p(s, r)}{p(s)p(r)}$$

其中, $p(s) = \sum_{r \in R} p(s, r)$ 和 $p(r) = \sum_{s \in S} p(s, r)$ 分别是变量 S 和变量 R 的边缘分布。

对于用粗糙集理论处理的决策表中的属性约简,需要考虑的问题是哪个属性对于决策属性来说是重要的。在决策表中增加某个属性所引起的互信息的变化可以作为该属性重要性的度量。所以可以通过互信息的增加值计算来评价各个条件属性的重要性。

给定一个决策信息系统, $S = \langle U, R, V, f \rangle$, $R = C \cup D$ 是非空有限的属性集合, C 是条件属性集, D 是决策属性。有 $B \subset C$, 添加一个属性 $a \in C$ 到 B 里去, 则获取的互信息增加量 gain 为:

$$\text{gain} = MI(B \cup \{a\}; D) - MI(B, D)$$

计算每个条件属性的互信息增值 gain , 具有最大 gain 值的条件属性就是对决策属性最重要的。

我们已经知道, 属性选择需要挑选有较高的样本覆盖率和占有率的属性, 所以应该选择包含尽可能多的样本数目的属性, 而当待选的多个属性包含相同数目的样本时, 应该选择其中含有尽可能少的不同属性取值的属性^[12]。

通过研究, 发现 $\text{POS}_B(D) / \text{IND}(\{B, D\})$ 中成员

包含对象的最大数目和样本覆盖率及属性重要性有关。正域 $\text{POS}_B(D)$ 的基数 $\text{card}(\text{POS}_B(D))$ 实际就等同于一致样本的数目, $\max \text{card}(\text{POS}_B(D))/\text{IND}(\{B, D\})$ 表示正域中包含的最大不可分辨类中具有的对象数目。这个最大不可分辨类是在属性选择过程中由正域 $\text{POS}_B(D)$ 被属性集分类而得到的。如果一个属性具有较多的不同属性值, 那么通常其相应也会得到较多的分类子集。所以用这种方法来进行属性选择是可行的。选择一个属性 $\{a\}$ 并将其加入到属性子集 B 中去, 相比加入其他属性, 如果有 $\text{card}(\text{POS}_{(B \cup \{a\})}(D))$ 增长更快并且 $\max \text{card}(\text{POS}_{(B \cup \{a\})}(D))/\text{IND}(\{B \cup \{a\}, D\})$ 更大的话, 那么属性 $\{a\}$ 就是最重要的, 就可以将属性 $\{a\}$ 加入属性子集。当遇到有两个或两个以上待选属性具有相同的 $\max \text{card}$ 值时, 就选择使得互信息增加最大的那个属性。

2.2 算法描述

令 B 是一个已选择的条件属性集, C 是条件属性集, 并且 $B \subseteq C$, D 是决策属性, L 表示还未被选择的条件属性集, $T(L)$ 是 L 的幂集, $T_i(L) (i = 1, 2, \dots, m)$ 是 L 的第 i 阶幂子集。 U 表示样本空间, X 表示不一致样本集, EXPECT 是用户定义的某种准确阈值, RED 表示约简的属性集合。

步骤 1: 令 $B = \text{CORE}(C, D)$, $L = C - \text{CORE}(C, D)$, $k = 0$; 令 $X = U - \text{POS}_B(D)$ 。

步骤 2: 如果 $k \geq \text{EXPECT}$, $k = r(C, D) = |\text{POS}_C(D)| / |U|$, 则结束。或者如果 $\text{POS}_B(D) = \text{POS}_C(D)$, 则结束。

步骤 3: 令 $i = 1$, $\text{flag} = 0$, $Z = A = \emptyset$, $Y = T_i(L)$ 。

步骤 4: 任意选择 $y \in Y$, 令 $A = B \cup y$, 如果 $\text{POS}_A(D) = \text{POS}_C(D)$, 那么则 { 如果 $\text{flag} = 0$, 则令 $Z = A$, $\text{flag} = 1$; } 否则如果 $\text{card}(U/Z) > \text{card}(U/A)$, 则令 $Z = A$ 。

步骤 5: 如果 $\text{flag} = 1$, 则计算 $V_y = \text{card}(\text{POS}_{(B \cup y)}(D))$ 和 $\max \text{card}$ 值。

$M_y = \max \text{card}(\text{POS}_{(B \cup \{y\})}(D))/\text{IND}(\{B \cup \{y\}, D\})$

步骤 6: 选择最好的属性 y , 也就是具有最大值 $V_y \times M_y$, 令 $Y = Y - y$, $B = B \cup y$, 并且令 $\text{RED} = B$ 。

步骤 7: 如果出现不同属性具有相同的 $\max \text{card}$ 值, 则计算 $\text{MI}(D | \{a_j\})$, 令 $M_j = \max(\text{MI}(D | \{a_j\}))$, 选择对于 D 具有最大互信息获得的 $\{a_j\}$, 且令 $\text{RED} = \text{RED} \cup \{a_j\}$ 。

步骤 8: 令 $i = i + 1$, 转到步骤 1。

这个算法的复杂度包含了 4 个方面的计算负载:

(1) 计算属性核 $\text{core}(C, D)$ 需要计算 $|C|$ 次, 所以这部分的时间复杂度是 $o(|C|)$;

(2) 计算正域需要计算 $(|C| + |C - 1| + \dots + 1) = |C| \times (|C| + 1)/2$ 次, 这部分的时间复杂度是 $o(|C|^2)$;

(3) 计算属性重要性 $r(C, D)$ 需计算 $(|C| \parallel U|^2)$ 次, 这部分的时间复杂度是 $(|C| \parallel U|^2)$;

(4) 计算互信息增值的时间需要 $(|P| \parallel U|^2)$, 其中 P 是约简以后决策信息表中还剩下的对于决策属性具有相同的 $\max \text{card}$ 值的属性集合。最坏的情况就是 $|P| \approx |C|$, 即这部分最坏的时间复杂度是 $o(|C| \parallel U|^2)$ 。所以, 在最坏的情况下, 这个算法的整个时间复杂度是 $o(|C|^2) \times o(|C| \parallel U|^2) = o(|C|^3 \times |U|^2)$ 。

2.3 实验示例

下面给出一个示例来说明这个算法。表 1 是一个决策表, $\{a_1, a_2, a_3, a_4, a_5\}$ 是条件属性集, $\{d\}$ 是决策属性。

表 1 决策信息表

U	a_1	a_2	a_3	a_4	a_5	d
x_1	1	1	3	2	1	2
x_2	1	1	3	1	1	2
x_3	1	3	1	1	1	3
x_4	1	3	3	2	2	1
x_5	2	2	1	1	2	3
x_6	2	2	2	1	1	3
x_7	2	2	3	2	1	2
x_8	1	3	2	1	1	3
x_9	2	3	3	3	3	2

$U/\{a_2\} = \{\{x_1, x_2\}, \{x_5, x_6, x_7\}, \{x_3, x_4, x_8, x_9\}\}$

$U/\{d\} = \{\{x_4\}, \{x_1, x_2, x_7, x_9\}, \{x_3, x_5, x_6, x_8\}\}$

可见 $\{d\}$ 的 $\{a_2\}$ 正域: $\text{POS}_{a_2}(d) = \{x_1, x_2\}$ 。根据上面给出的算法步骤, 假设 $B = \text{CORE} = \{a_2\}$, 那么 $L = \{a_1, a_3, a_4, a_5\}$, $X = \{x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$ 。设定准确阈值 $\text{EXPECT} = 1$, 则终止条件即是 $k \geq 1$ 。有 $k = r(C, D) = |\text{POS}_C(D)| / |U| = 2/9 < 1$, 所以 B 并不是约简集, 需要继续选择条件属性加入 B 。下一步待选的条件属性有 $\{a_1\}, \{a_3\}, \{a_4\}$ 和 $\{a_5\}$ 。表 2, 表 3, 表 4, 表 5 分别给出了加入 $\{a_1\}, \{a_3\}, \{a_4\}$ 及 $\{a_5\}$ 到 B 的结果。

表 2 选择属性 $\{a_1\}$

U	a_2	a_1	d
x_3	3	1	3
x_4	3	1	1
x_5	2	2	3
x_6	2	2	3
x_7	2	2	2
x_8	3	1	3
x_9	3	2	2

表 3 选择属性 $\{a_3\}$

U	a_2	a_1	d
x_3	3	1	3
x_4	3	3	1
x_5	2	1	3
x_6	2	2	3
x_7	2	3	2
x_8	3	2	3
x_9	3	3	2

表 4 选择属性 $\{a_4\}$

U	a_2	a_1	d
x_3	3	1	3
x_4	3	2	1
x_5	2	1	3
x_6	2	1	3
x_7	2	2	2
x_8	3	1	3
x_9	3	3	2

表 5 选择属性 $\{a_5\}$

U	a_2	a_1	d
x_3	3	1	3
x_4	3	2	1
x_5	2	2	3
x_6	2	1	3
x_7	2	1	2
x_8	3	1	3
x_9	3	3	2

根据表 2 至表 5,可以得到添加各个属性后得到的等价分类以及各个相关量的计算值。

$$\begin{aligned} U/d &= \{\{x_3, x_5, x_6, x_8\}, \{x_4\}, \{x_7, x_9\}\}; \\ U/\{a_1, a_2\} &= \{\{x_3, x_4, x_8\}, \{x_5, x_6, x_7\}, \{x_9\}\}; \end{aligned}$$

$$\begin{aligned} \text{POS}_{|a_1, a_2|}(d) &= \{x_9\}, \\ \max_ \text{card}(\text{POS}_{|a_1, a_2|}(d)/\{a_1, a_2, d\}) &= 1; \\ U/\{a_2, a_3\} &= \{\{x_3\}, \{x_4, x_9\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}, \\ \text{POS}_{|a_2, a_3|}(d) &= \{x_3, x_5, x_6, x_7, x_8\}, \\ \max_ \text{card}(\text{POS}_{|a_2, a_3|}(d)/\{a_2, a_3, d\}) &= 1; \\ U/\{a_2, a_4\} &= \{\{x_3, x_8\}, \{x_4\}, \{x_5, x_6\}, \{x_7\}, \{x_9\}\}, \\ \text{POS}_{|a_2, a_4|}(d) &= \{x_3, x_4, x_5, x_6, x_7, x_8, x_9\}, \\ \max_ \text{card}(\text{POS}_{|a_2, a_4|}(d)/\{a_2, a_4, d\}) &= |\{x_3, x_8\}| = |\{x_5, x_6\}| = 2; \\ U/\{a_2, a_5\} &= \{\{x_3, x_8\}, \{x_4\}, \{x_5\}, \{x_6, x_7\}, \{x_9\}\}, \\ \text{POS}_{|a_2, a_5|}(d) &= \{x_3, x_4, x_5, x_8, x_9\}, \\ \max_ \text{card}(\text{POS}_{|a_2, a_5|}(d)/\{a_2, a_5, d\}) &= |\{x_3, x_8\}| = 2 \\ \text{MI}(\{a_2, a_4\}; d) &= 0.258; \\ \text{MI}(\{a_2, a_5\}; d) &= 0.172. \end{aligned}$$

由以上算法演示过程可以得知,选择添加属性 $\{a_1\}$ 并不能减少不一致样本的数目,但是选择添加属性 $\{a_3\}, \{a_4\}$ 或 $\{a_5\}$ 可以明显减少不一致的样本数目。当选择添加属性 $\{a_4\}$ 时,所有样本都变为一致了。根据我们的算法,具有最大基数的不可分辨类是从 $\text{POS}_{|a_2, a_4|}(d)/\{a_2, a_4, d\}$ 和 $\text{POS}_{|a_2, a_5|}(d)/\{a_2, a_5, d\}$ 中得到的,两者具有相同的 $\max_ \text{card}$ 值,所以要继续分别计算添加属性 $\{a_4\}$ 以及添加属性 $\{a_5\}$ 的互信息增值,然后选择其中具有较大互信息增值的属性 $\{a_4\}$ 添加到属性约简集中。最后,选择的约简属性集是 $\{a_2, a_4\}$ 且此时 $\text{POS}_{|a_2, a_4|}(d)/U = 1$,算法终止。

2.4 算法分析

在粗糙集决策信息系统的属性约简过程中,使用启发式的属性搜索方法显然要比其他穷尽式的搜索方法好,其搜索时间快,计算复杂度小。由于首先得到属性核,这样就避免了对属性之间随机组合情况的搜索,又由于对属性重要性进行了排序,就避免了对超出最小属性约简个数的属性组合情况的搜索,因此可以提高求解速度,并可以得到最小属性约简结果。

为了证实这个算法的有效性,从 UCI 机器学习知识库中选择几个数据库 Monk1, Heart Disease, Mushroom, Breast - Cancer 和 Slope Collapse 来验证一下。使用我们的算法和 Rosetta 软件中包含的动态约简算法以及标准遗传算法来处理这些数据库,从而得出比较结果。比较的结果如表 6 所示。

表 6 对 UCI 中五个数据库使用不同属性约简算法处理的比较

数据集	样本数 n	属性数 n	约简集属性个数 n		
			遗传算法	动态约简算法	新算法
Monk1	124	6	3 或 4	3 或 4	3
Heart Disease	294	14	3 或 4 或 5	3 或 4 或 5 或 6	3
Mushroom	8124	22	5 或 6 或 7	5 或 6 或 7 或 8	5
Breast - Cancer	699	10	5 或 6 或 7	5 或 6 或 7	5
Slope Collapse	3436	24	9 或 10 或 11	9 或 10 或 11 或 12	9

通过实验和比较可知,使用标准遗传算法和动态约简算法都得到了好几个属性约简集,无法得知哪个属性约简集是最好的。而我们提出的约简算法仅得到具有最少属性个数的属性约简集,有效且提高了约简速度。

3 结束语

启发式属性约简算法由于首先是从属性核开始添加选择的属性,这样就避免了对属性之间随机组合情况的搜索,又由于每一步骤的属性选择时,都通过不一致计数和互信息增量的计算对待选属性的重要性进行了排序,就避免了对超出最小属性约简个数的属性组合情况的搜索,因此可以提高求解速度,并可以得到最小属性约简结果。此方法适合于有多个条件属性的情况,条件属性与属性核的选择需要根据具体情况来确定。

参考文献:

[1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science,1982,11(5):341-356.
[2] Pawlak Z. Rough set - Theoretical Aspects of Reasoning about Data[M]. Dordrecht:Kluwer Academic Publishers,1991.
[3] Pawlak Z, Slowinski R. Rough set approach to multiattribute

decision analysis[J]. Invited Review, European Journal of Operational Research,1994,72:443-459.

[4] Tseng T L, Huang C C. Rough set - based approach to feature selection in customer relationship management[J]. Omega, 2007,35:365-383.
[5] 梁吉业,曲开社,徐宗本. 信息系统的属性约简[J]. 系统工程理论与实践,2001(12):76-80.
[6] 芦晓红,陈世权,吴今培. 基于可辨识矩阵的启发式属性约简方法及其应用[J]. 计算机工程,2003,29(1):56-58.
[7] 徐章艳,杨炳儒. 一个基于决策表的快速属性约简算法[J]. 小型微型计算机系统,2006,27(5):858-861.
[8] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of Polish Academy of Sciences, 1985,33: 693-696.
[9] Jelonek J, Krawiec K, Slowinski R. Rough set reduction of attributes and their domains for neural networks[J]. Computational Intelligence,1995,11(2):339-347.
[10] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001.
[11] 王国胤,于洪,杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报,2002,25(7):759-766.
[12] Ning Z, Juzhen D, Setsuo O. Using Rough sets with Heuristics for Feature selection[J]. J. Intell. Inf. Syst, 2001, 16: 199-214.

(上接第 15 页)

[7] 刘峰,顾君忠. 元数据管理应用系统的设计与实现[J]. 计算机工程,2009,35(11):29-31.
[8] 沈盛斌,刘恺,姚文龙. 基于 Web 服务的空间元数据管理平台研究[J]. 测绘与空间地理信息,2009,32(4):13-15.
[9] 谢福成,王备战,史亮,等. 基于银行数据仓库的元数据管理系统[J]. 计算机工程,2009,35(9):79-81.
[10] 蔡昭权,卢庆武,郑宗晖,等. 基于元数据的快速开发平台

设计与实现[J]. 计算机工程,2009,35(9):60-62.

[11] 徐财江,陈和平,陈志荣. 土地利用现状数据元数据管理系统的设计与实现[C]//2006 年中国土地学会学术年会论文集. 重庆:[出版者不详],2006:707-713.
[12] 何清林,杨森,徐泽同. 基于元数据和 Web Service 中间件的分布式资源库集成[J]. 计算机工程与设计,2009,30(9):2201-2203.