Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

A CLASSIFICATION ALGORITHM BASED ON MULTI-RELATION DOMAIN KNOWLEDGE

XUE-GANG HU, XIE-FEI HU, DE-XING WANG, DONG-YAN ZHANG, CHUN-LING HU

School of Computer and Information, Hefei University of Technology, Hefei 230009, China
E-MAIL: jsjxhuxg@hfut.edu.cn, huxiefei@etang.com, wangdexing198706@yahoo.com.cn

Abstract:

Recently, integrating domain knowledge in knowledge discovery has engaged many experts' attention. Various kinds of domain knowledge can take on a number of different forms and have a great diversity of effects in knowledge discovery in database. Concept lattice, which has a complete form in knowledge representation, can represent the complex relations among the objects in the world. In the paper, incorporating multi-relation domain knowledge in knowledge discovery is proposed and a classification algorithm CS_MRDK that using concept lattice for discovering the closest relationship among attributes is presented, which shows the good performance of using multi-relation domain knowledge in knowledge discovery.

Keywords:

KDD; Domain knowledge; Concept lattice; Classification

1. Introduction

With the development of information technology and the explosive growth of the size of databases, knowledge discovery in database (KDD) are facing new problems such as focusing search to relevant portion of data, making the discovered patterns more meaningful and so on. Additional knowledge, called domain knowledge (DK), is used to help the discovery process such as data pre-processing, finding the strong relevant attributes, generalizing the concepts to more interesting hierarchies, guiding the discovery process to extract more useful rules, interpreting the discovered results and making the results be more understandable to the end user and so on, so as to make the discovery process more efficiently and effectively. However, the research and application referring to this research field are at a primary stage, thus deserve our exploring and further research. KDD based on DK is a newly promising research field.

An algorithm for inducing classification rules is presented in this paper, which can introduce inner relationship of attributes into rules by domain knowledge with multi-relations. The rest of the paper is organized as follows. Section 2 introduces the relevant work of domain knowledge. Section 3 describes multi-relation domain knowledge and gives definitions about concept lattice that can be used for representing domain knowledge. Section 4 first gives three theorems for mining classification rules based on extended concept lattice and then presents an algorithm CS_MRDK for mining classification rules using multi-relation domain knowledge. Section 5 uses an example to illustrate the algorithm and section 6 analyses the algorithm. Section 7 reaches a conclusion and outlines the directions for further research.

2. Domain knowledge in knowledge discovery

Domain knowledge can be defined as any additional information that is not explicitly presented in database. It can assist discovery by focusing search for interesting knowledge and rule out unvalued discovery [1], [2]. In some of the references, there is another phrase, i.e., background knowledge. According to the definition, we may find that there is no explicit difference between these two terms and in fact it is difficult to distinguish one from the other. So we identify them with the same one.

It was in 1991 that Frawley proposed that domain knowledge can be used in all aspects of automated discovery and the discovery process should be guided by proper domain knowledge [1]. In the nearest ten years, a series of researches referring to representation of domain knowledge and the corresponding mechanism and methodology of using domain knowledge in knowledge discovery process have been explored at home and abroad. Owrang O. M. M. represents domain knowledge in the form of X implies Y, and by using this kind of available domain knowledge, the author minimizes search time by reducing the size of databases, optimizing the hypothesis that represent the knowledge to be discovered, and optimizing queries that are used to prove the hypothesis [2]. Anand, Sarabjot S. and Bell, David A. et al. define three classes of domain knowledge that are hierarchical generalization trees, attribute relationship rules and

0-7803-9091-1/05/$20.00 ©2005 IEEE

environment-based constraints and discuss how each class of domain knowledge is incorporated into the discovering process [3]. Jiawei Han et al. represent domain knowledge in the form of concept trees and put forward Attribute-Oriented Induction (AOI) method that extracts generalized data from actual data in databases, the key of which is the attribute-oriented concept tree ascension for generalization, which substantially reduces the computational complexity of the discovery processes and improves the quality of the discovered results [4], [5], [6]. Suk-Chung Yoon, Lawrence J. Henschen and E. K. Park et al. take advantage of domain knowledge about the contents of the database and utilize three types of domain knowledge for semantic query optimization, namely, inter-field domain knowledge, category domain knowledge and correlation domain knowledge [7]. Carsten Pohle proposes to employ formalized domain knowledge and plans to apply ontology for construction of intelligent data mining environments [8]. Heekyoung Seo and Jaeyoung Yang et al. present a method of building intelligent systems for mining information extraction rules from semi-structured web pages by using domain knowledge [9].

3. Multi-relation domain knowledge

Domain knowledge may originate from many sources, such as data dictionary, end users or domain experts, and discovered knowledge etc. The model that can represent knowledge effectively is an important research issue in knowledge discovery in database.

In the previous research, hierarchy domain knowledge usually represented as concept trees that can reflect the hierarchy relationship of attributes. Users obtain the generalized description of the actual data by using the attribute-oriented concept tree ascension. However, the attribute-oriented ascension simply replaces low-level tuples by high ones and requires users' explicit preferences if users want to change the different concept hierarchies constructed on the same attribute due to the limitation of trees, which is restricted by the subjective factor of the user so that constrain the quality of the discovered knowledge.

In target dataset, the values of some attributes may be low-level descriptions in the form of symbols and may have no explicit relationship with others to all appearances. This incompleteness of the target dataset leads to limited information and makes it hard to find the potential useful patterns for the end users. Domain experts or users are needed to provide relevant information for resolving these problems. While in this paper, we build the relationship among these symbols based on domain knowledge or background knowledge, which is in the form of multi-relation tables and can be used to guide the knowledge discovery process automatically.

Concept lattice [10], which is a complete form of knowledge representation, can be used to represent the complex relationships among objects in the world. Discovering based on multi-relation domain knowledge represented by concept lattice helps to guide the discovery process to achieve more meaningful descriptions and find the closest relationship at a high level automatically.

Concept Lattice, also called Galois Lattice, was proposed by R.Wille in 1982 and is built on binary relationship between the intensions and extensions of concepts and can be used to represent the relationships between the generalization and the specialization of concepts. Here gives the corresponding definitions of concept lattice.

Definition 1: A *context* is defined as a triple T = (O, D, R), where O is a set of objects, D is a set of attributes, and R is a binary relationship between O and D. In the context, there is a unique ordered set that describes the structure of inherent lattice, which defines natural groupings and relationship descriptions among the objects and their attributes. This structure is also known as *Concept Lattice* or *Galois Lattice* (*GCL*).

Definition 2: A couple such as C = (A, B) derived from the context (O, D, R) is called a *basic concept*, where $A \subseteq O$, $B \subseteq D$, $B' = \{g \in O \mid \forall m \in B, gRm\} = A$. A is called the *extension* of the concept, and B is called the *intension*. If there are two concepts $C_1 = (A, B_1)$ and $C_2 = (A, B_2)$, then B_1, B_2 are called *equivalent basic intensions*, denote as $Equ(A) = \{B_i \mid B_i \subseteq A' \text{ and } B_i' = A\}$. Define C= (A, Equ(A)) as *extended concept*, or simply called *concept*. The set of extended concepts constitute a complete lattice called *Extended Concept Lattice*, denoted as *ECL* [11], [12].

In particular, the concept containing all objects of C is called *full concept*, and the concept containing none of objects of C is called *empty concept*.

Definition 3: In a ECL, partial order relation"<*" between concept $C_1=(A_1, B_1)$ and $C_2=(A_2, B_2)$ is defined as $C_1 <^* C_2$ if and only if $A_1 \subseteq A_2$, C_1 is the *sub-concept* of C_2, and C_2 is the *super-concept* of C_1. If $C_1 <^* C_2$, and there isn't a concept C such that $C_1 <^* C <^* C_2$, the relation between C_1 and C_2 is called *direct-sub-concept--direct-super-concept*, denote as $C_1 < C_2$.

4. A classification algorithm based on multi-relation domain knowledge

ECL that is used as the representation of multi-relation domain knowledge for describing the information deeply hide among the attributes also can be used in discovering

various rules in target dataset including classification rules, association rules and so on. In fact, the target dataset can be represented in various forms whereas is represented by ECL in this paper.

4.1. Mining classification rules in ECL

In the following, several theorems for mining classification rules based on ECL are presented. We assume that the class attribute is T with the range $V(T)=\{T_1, T_2, \dots, T_k\}$.

Theorem 1: For concept $C_1=(A_1, \{T_1\} \cup B_1)$, if there exists $B_{1i} \in B_1$ while $T_1 \notin B_{1i}$, then the classification rules as $B_{1i} \Rightarrow T_1$ ($B_{1i} \in B_1$ is the basic intension) can be drawn.

Proof: There is the equivalent rule $T_1 \Leftrightarrow B_1$, and for $T_1 \notin B_{1i}$, we can draw $B_{1i} \Rightarrow T_1$ as a classification rule.

Theorem 2: If There are two concepts $C_1=(A_1, \{T_1\} \cup B_1)$ and $C_2=(A_2, B_2)$ in ECL, where $C_2<C_1$, $B_2 \neq \varnothing$, then the classification rules as $B_{2i} \Rightarrow T_1$ ($B_{2i} \in B_2$ is the basic intension) can be drawn.

Proof: Because $C_2<^*C_1$, then $A_2 \subset A_1$, and A_1RT_1 is true, so A_2RT_1 is true, so has the classification rule $B_{2i} \Rightarrow T_1$.

Theorem 3: If There are two concepts $C_1=(A_1, \{T_1\} \cup B_1)$ and $C_2=(A_2, B_2)$ in ECL, where $C_2<^*C_1$, $B_2 \neq \varnothing$, then the classification rules as $B_{2i} \Rightarrow T_1$ ($B_{2i} \in B_2$ is the basic intension) can be drawn.

Proof: Referring to Theorem 2.

4.2. A classification algorithm

The main idea of algorithm CS_MRDK is that for the attributes in target dataset that have domain knowledge in the form of multi-relation tables, importing the relevant domain knowledge according to the disparate attribute values in the target dataset and finding the closest descriptions that express the relationship in the domain knowledge according to class label attribute; inserting the concepts that reflect the closest descriptions into the lattice that constructed based on the target dataset, so as to construct a new lattice and discovering classification rules based on the new lattice.

Algorithm CS_MRDK:

//Discovering classification rules using MRDK.

//Assume that c is the class label attribute.

Input: ⑴the target dataset D;

⑵the domain knowledge datasets which are in the form of multi-relation tables

Output: the classification rules of the target dataset

⑴If there isn't additional domain knowledge for every attribute in D , then construct lattice1 according to D, and discover the classification rules based on lattice1;

Else for each attribute $a \in A$, if there exists relevant domain knowledge of a, then

①Constructing lattice1 according to D, and getting the following data structure simultaneously:

list(a,i): the disparate attribute values of a

list(a,i).object: the corresponding objects for each list(a,i)

extension(c_i): the corresponding objects for each attribute value c_i of c

list(a,extension(c_i)): the corresponding values of a for each extension(c_i)

②Importing relevant domain knowledge according to list(a,i);

Constructing lattice2;

⑵//Finding the closest descriptions based on lattice2

①Getting all subsets m of list(a,extension(c_i)) such that element_number(m)> thresh;

Finding all concept x in lattice2 that has extension m;

Putting x into DK(a);

② Modifying the extension of concepts in DK(a) according to list(a,i).object;

Inserting the newly produced concepts into lattice1 and constructing lattice3;

③Discovering classification rules based on lattice3;

⑶Showing the results to the users. If the users are not satisfied with the results, then go to ⑴.

5. Example

Let table 1 is the target dataset D, and g is the class label attribute. The learning task is to finding the classification rules for D.

Table1: target dataset D

n	u	s	g
n1	u1	s1	g2
n2	u2	s3	g3
n3	u3	s3	g3
n4	u2	s3	g3
n5	u4	s1	g2
n6	u5	s3	g1
n7	u6	s1	g2
n8	u7	s2	g3
n9	u8	s2	g2

There isn't any relationship between attribute u and attribute g to all appearances. However, if we have addition knowledge about attribute u, we'll discover the inner

relationship between these two attributes that may hide deeply. Viewing the relation in table1 as a context and there is relevant domain knowledge for attribute u. The algorithm CS_MRDK is done as follows:

⑴There is additional domain knowledge for attribute u, then

①Constructing lattice1 according to D, as shown in figure1 and getting the following data values simultaneously, as shown in table2 and table3:

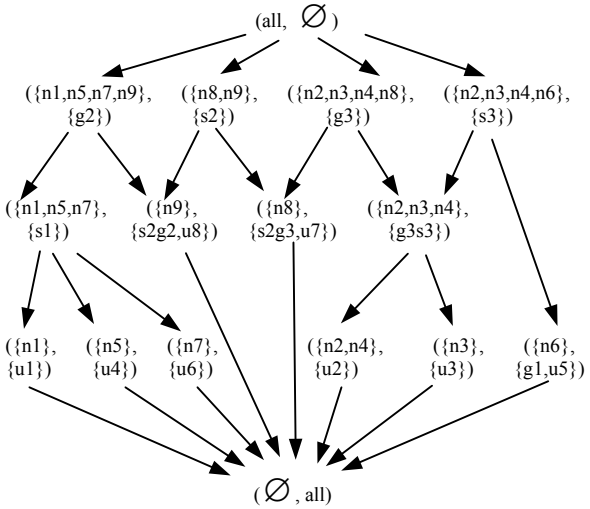

Figure 1. The initial lattice (lattice1)

Table2

i	list(u,i)	list(u,i).object
1	u1	{n1}
2	u2	{n2,n4}
3	u3	{n3}
4	u4	{n5}
5	u5	{n6}
6	u6	{n7}
7	u7	{n8}
8	u8	{n9}

Table3

i	extension(g_i)	list(u,extension(g_i))
1	{n6}	{u5}
2	{n1,n5,n7,n9}	{u1,u4,u6,u8}
3	{n2,n3,n4,n8}	{u2,u3,u7}

② Importing relevant domain knowledge according to list(u,i) as shown in Table4, and constructing lattice2, as shown in figure2.

⑵//Finding the closest descriptions based on lattice2

//The default threshold value is 1.

①For list(u,extension(g1))={u5}, element_number(m)$\leq$1.

For list(u,extension(g2))= {u1,u4,u6,u8}, using {u1, u4, u6, u8} or subset of {u1,u4,u6,u8} as the extension of a concept, the corresponding concepts in lattice2 are ({u1, u4, u8},{l1}), ({u1, u4},{t1}).

For list(u,extension(g3))= {u2,u3,u7}, using {u2,u3,u7} or subset of {u2,u3,u7} as the extension of a concept, the corresponding concepts in lattice2 is ({u2, u3, u7},{l3}), ({u2, u7},{l3t2, c2}).

Table4: relevant domain knowledge

u	c	l	t
u1	c1	l1	t1
u2	c2	l3	t2
u3	c3	l3	t3
u4	c4	l1	t1
u5	c4	l2	t3
u6	c5	l2	t3
u7	c2	l3	t2
u8	c6	l1	t2

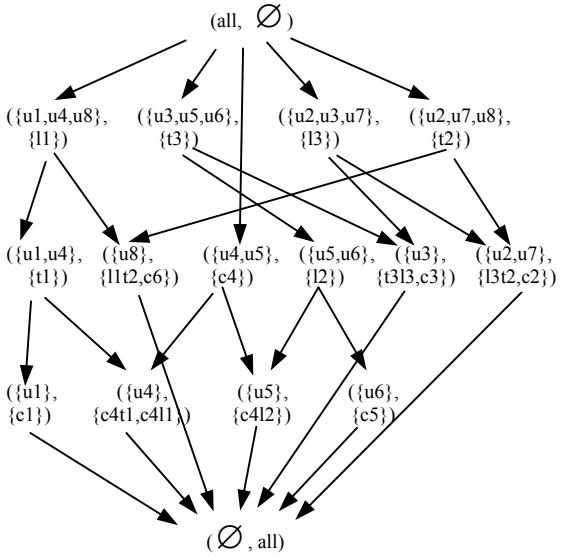

Figure 2.The lattice of relevant domain knowledge(lattice2)

Therefore, DK(u) = {({u2, u3, u7},{l3}), ({u2, u7},{l3t2, c2}), ({u1, u4, u8},{l1}), ({u1, u4},{t1})}.

②Modifying the extension of concepts in DK(u), the result is ({n2, n3, n4, n8},{l3}), ({n2, n4, n8},{l3t2, c2}), ({n1, n5, n9},{l1}), ({n1, n5},{t1}).

Inserting the newly produced concepts into lattice1 and constructing lattice3, as shown in figure3.

③Inducing classification rules based on lattice3, such as: u5$\Rightarrow$g1, s1$\Rightarrow$g2, u1$\Rightarrow$g2, u4$\Rightarrow$g2, u6$\Rightarrow$g2, u8$\Rightarrow$g2, l1$\Rightarrow$g2, t1$\Rightarrow$g2, u2$\Rightarrow$g3, u3$\Rightarrow$g3, u7$\Rightarrow$g3, l3$\Rightarrow$g3, c2$\Rightarrow$g3.

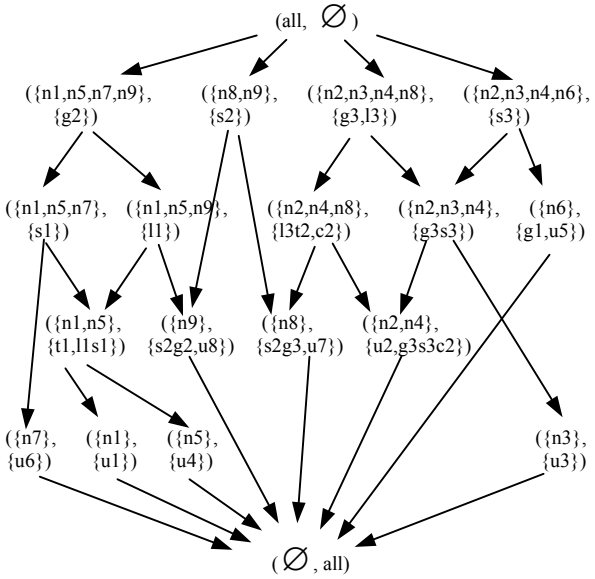

Figure 3. The new lattice(lattice3)

6. Algorithm analysis and comparison

If traditional classification algorithm based on extended concept lattice is used without taking domain knowledge into account, we will get lattice shown in figure1. Extracting classification rules from such lattice, we will get the following classification rules: u5 $\Rightarrow$ g1, s1 $\Rightarrow$ g2, u1 $\Rightarrow$ g2, u4 $\Rightarrow$ g2, u6 $\Rightarrow$ g2, u8 $\Rightarrow$ g2, u2 $\Rightarrow$ g3, u3 $\Rightarrow$ g3, u7 $\Rightarrow$ g3. On the one hand, all these rules are included in the result of algorithm CS_MRDK, thus shows the completeness of CS_MRDK. On the other hand, rules l1 $\Rightarrow$ g2, t1 $\Rightarrow$ g2, l3 $\Rightarrow$ g3, c2 $\Rightarrow$ g3 are not included which reflect the implicit relationship among the features of some condition attributes and the decision attribute. That is to say, discovering based on multi-relation domain knowledge helps to find the implicit and potential useful patterns, which is congruous to the intension of knowledge discovery in database.

If multi-relation domain knowledge is taken into account when inducing classification rules whereas simply amalgamating the target dataset with the relevant domain knowledge instead of discovering the closest relationship based on extended concept lattice, the dataset will be expanded in dimension. When the size of relevant domain knowledge is small while the size of target dataset is large, the amalgamating process means that the target dataset is expanded with many dimensions that have the same attribute values. Knowledge discovery on such kind of dataset is inefficient and time consuming obviously. Algorithm CS_MRDK imports the relevant domain knowledge according to the attribute values in target dataset, finds the closest descriptions of these attributes, and then incorporates these descriptions that are small in size into the target dataset. So this algorithm neither needs to amalgamate data from the target dataset and domain knowledge datasets nor needs to deal with the large amount of amalgamated data. Therefore, it is efficient and practical.

7. Conclusions

This paper incorporates multi-relation domain knowledge that can be used to find the inner relationship of attributes into knowledge discovery and utilizes concept lattice for describing the closest relationship among attributes. A classification algorithm is also presented, which demonstrates the superiority of using multi-relation domain knowledge represented by concept lattice. Furthermore, there are other types of domain knowledge representations. The consummate mechanism and methodology for discovering knowledge deserve our further research.

Acknowledgements

The paper is supported by the Natural Science Foundation of Anhui Province under Grant No. 050420207 and Research and Development Foundation of Hefei University of Technology under Grant No. 050504F.

References

[1] W.J.Frawley, G.Piatetsky-Shapiro, C.J.Matheus, Knowledge Discovery in Database: An Overview, In: G. Piatetsky-Shapiro, W.J.Frawley, eds. Knowledge Discovery in Databases, Menlo Park, California: AAAI Press / The MIT Press, pp. 1-27, 1991.

[2] Owrang O. M. M., Optimization of Knowledge Discovery Process Using Domain Knowledge, Proceeding of Intelligent Information Systems, pp. 428-433, 1997.

[3] Anand, Sarabjot S., Bell, David A., Hughes, John G., The Role of Domain Knowledge in Data Mining, Proceeding of the Fourth International Conference on Information and Knowledge Management, pp. 37-42, 1995.

[4] Yandong Cai, Nick Cercone, and Jiawei Han, Attribute-Oriented Induction in Relation Databases, In: G. Piatetsky-Shapiro, W.J.Frawley, eds. Knowledge Discovery in Databases, Menlo Park, California: AAAI Press / The MIT Press, pp. 213-228, 1991.

[5] Jiawei Han, Yandong Cai, and Nick Cercone, Knowledge Discovery in Databases: An Attribute-Oriented Approach, Proceeding of the 18th VLDB Conference, 1992.

[6] Jiawei Han, Mining Knowledge at Multiple Concept Levels, Proceeding of the Fourth International Conference on Information and Knowledge Management, pp. 19-24, 1995.

[7] Suk-Chung Yoon, Lawrence J. Henschen, E. K. Park, Sam Makki, Using Domain Knowledge in Knowledge Discovery, Proceeding of the Eighth International Conference on Information and Knowledge Management, pp. 243-250, November 1999.

[8] Carsten Pohle, Integrating and Updating Domain Knowledge with Data Mining, Proceeding of the VLDB 2003 PhD Workshop, Berlin, Germany, September 12-13, 2003.

[9] Keekyoung Seo, Jaeyoung Yang, Joongmin Choi, Building Intelligent Systems for Mining Information Extraction Rules from Web Pages by Using Domain Knowledge, Proceeding of IEEE International Symposium on Industrial Electronics, pp. 322-327, 2001.

[10] R.Wille, Restructuring lattice theory: An approach based on hierarchies on concepts, Proceeding of the NATO Advanced Study Institute, Banff, pp. 445-470, 1982.

[11] Xuegang Hu, The Research of Models on Knowledge Discovery in Databases, [PH.D Dissertation], Hefei university of Technology, Hefei 2000, Anhui, China (in Chinese).

[12] De-Xing Wang, Xue-Gang Hu, Hao Wang, The Research on Model of Mining Association Rules Based on Quantitative Extended Concept Lattice, Proceeding of the First International Conference on Machine Learning and Cybernetics, Beijing, pp. 134-138, November 2002.