Incremental Concept Lattice Structure and Synchronous Calculation of Intent Reduction

[1]Zhang Shuiping, [2]He Zhenzhen

[1]*School of Economy and Management, Anhui University of Science and Technology, Anhui Province, China*
[2]*School of Economy and Management, Anhui University of Science and Technology, Anhui Province, China*

Corresponding author is Zhang Shuiping

Abstract
Concept lattice provides an effective means for data analysis and knowledge extraction, it has the characteristics of the completeness and accuracy. In this paper, the synchronous calculation method of the intent reduction is studied, the variation characteristics of intent reduction of the concept lattice nodes is demonstrated, when the dynamic attributes is inserted , and the Godin algorithm is improved, the incremental calculation method of intent reduction is proposed.

Keywords: INCREMENTAL CALCULATION, INTENT REDUCTION, THE GODIN ALGORITHM, EPITAXIAL REDUCTION, IREDIC ALGORITHM

1. Introduction

Concept lattice is a kind of concept hierarchy, which is established by the binary relationship between the data set objects and attributes, it reflects the generalization and specialization relationships among concepts vividly and succinctly(Ganter, B. et al, 1999). As the formal tool of data analysis and knowledge formalization, concept lattice has been successfully applied in many fields (Godin, R. et al, 1999; Godin, R. et al, 2009). It has been the research focus in this field that utilizing the concept lattice to extract the containing rules. Concept lattice is utilized to extract containing rules has been the research focus in the field.

The essence of Godin (Godin, R. et al, 1999) using algorithms is to do the exhaustive testing, when the rules of a node generated, each descriptor subset in power set of node descriptor set should be inspected. Zhi-Peng Xie, etc. (2001) proposed the definition of intent reduction, revealed the essence of the containing rule extraction clearly, and put forward the methods of using the intent reduction to extract no redundant containing rule set, this paper. In the last, containing rules of incremental extraction is proposed according to combining with the incremental updating of concept lattice, but the specific methods are not given.

In this paper, based on the research of incremental updating of concept lattice and calculation principle of intent reduction of nodes, it is proposed that the synchronous calculation method of intent reduction on the basis of incremental structure of concept lattice of inserting the attributes.

2. Related Definitions and Conclusions

Definition 1: a form of background: K: = (G, M, I) is made up of two sets G, M and the relationship I between the G and M. The elements of G are called objects, the elements of M called attributes. $(g, m)\in I$ shows object g has the attribute m.

Definition 2: Assume that A is a subset of the objects set G, and B is a subset of the attributes set M, Defining the export operator f, g are as follows:

f (A)={m∈M | g∈A. gIm}; g (B) : = {g∈G| m∈B, gIm}

Definition 3: Binary group (A, B) is a formal concept in the background of (G, M, l), among them, A, B are the subset of G,M respectively, and satisfy the f (A) = B, g(B) = A. Then, we call A is the extension of concept (A, B), B is the connotation of the concept (A, B).

Definition 4: for a given concept C = (A, B), if an attribute set D meet the following two conditions:

1) g (D) = g(B) = A, (epitaxial invariance)

2) (∀ F ⊂ D) g(F) ⊃ g(D) = A, (minimality)

then, D is called an intent reduction of the concept C.

According to the definition 4, we know that the intent reduction has the quality of epitaxial invariance, which means that the concept connotation and its intent reduction have the same extension; intent reduction also has the quality of minimality, which means that removing any attributes from intent reduction will lead to the expansion of extension. Meanwhile, a concept connotation may has more than one reduction. The family set of all the connotation reduction of concept C is called the intent reduction set, notes IRed (C). As to the solution of intent reduction, we have the following conclusions (Liu,B . et al, 1995).

Proposition 1: for any concept C = (A, B), and an attribute subset D ⊆ B, g (D) = g(B) = A, if and only if each parent concept in c: pC = (pA, pB) ,we have D ∩ (B - pB) ≠ Φ

By definition 4 and proposition 1, we can easily design calculating steps of intent reduction of a concept (Godin, R. et al, 1999; Liu, W.M. et al, 2012).

Algorithm: IRed (C)

Input concept C , output the corresponding intent reduction set IRed

Step 1: for the k parent concepts {pCi} 1 ≤ i ≤ k in the concept C, we have Tmpi = Int(C)–Int(pCi)

Step 2: cRed {∪1 ≤ i≤ k {mi}| mi Tmpi};

Step 3: calculate all the minimal value Min (cRed) of the partial order set (cRed), we have IRed = Min (cRed).

The above calculation is a kind of progressive calculation method which proposed in paper (Liu, Z.Q. et al, 2013). It is conducted after construction, in fact, the complexity of the above steps 3 solving minimal value function is very high. So far, there is no better solution about how evaluate minimum (maximum) of a partial order set. The synchronous calculation method that this paper put forward divided this problem into multi-step incremental calculation, which reduce the calculation complexity of intent reduction to a certain extent.

In the principle analysis of the article below, we consider the dual problem of intent reduction: epitaxial reduction, researches and proves the changing rule of the concept lattice nodes epitaxial reduction in the case of inserting object incremental building cases. And on this basis, the synchronous calculation method of epitaxial reduction is given based on Godin algorithm. Finally, according to the dual principle of concept lattice, we can get the conclusions about the intent reduction calculation in the case of inserting incremental building cases of attribute.

3. Solving Method

Define 5: for a given concept C = (A, B), if the object set D meet the following two conditions:

1) f (D) = f (A) = B,

2) (∀ F ⊂ D) f(F) ⊃ f(D)=A,

D is called an epitaxial reduction of concept C.

According to the dual principle of concept lattice, we can easily get the following conclusion by the proposition 1 (a necessary condition for epitaxial reduction):

Theorem 1: for any concept C = (O1, D1) and a subset O2 ⊆ O1, and f(O2) = f(O1) = D1, if and only if for every child concept pC=(O3, D3) , we have O2∩(O1–O3) ≠ Φ.

According to definition 5 and theorem 1, we know that the epitaxial reduction of a concept is associated with the extension of the concept's child nodes, and the epitaxial reduction of a concept (A, B) is noted EXTRED (A) (sometimes using EXTRED (A, B), the equivalent). In the process of building concept lattice incrementally, when inserting an object x, the extension of the old nodes, child nodes and extension of child nodes do not change, so there is no need to consider; For generated value, extension and child nodes also unchanged, also do not need to consider; the extension of the updated nodes and new nodes change. And the updated edge only occurs at updated nodes, generated value and its corresponding new nodes. So we need to consider the situation of reduction change of updated nodes and new nodes. Here, first of all, the proof of monotonicity of epitaxial reduction function is given.

Theorem 2: assume that (A, B) is updated node, {(Ai, Bi)} 1≤i≤k are k child nodes of (A,B) . Note the epitaxial reduction of (A,B) is EXTRED(A) before inserting a new object x, the updated epitaxial reduction is EXTRED (A ∪ {x}), then,

1) ∀ Ext ∈ EXTRED(A∪{x})–EXTRED(A), we have x ∈ Ext.

2) EXTRED(A) ⊆ EXTRED(A∪{x}), namely the epitaxial reduction function is monotone increasing.

It is proved as follows:

1) as long as we can prove that arbitrary Ext∈EXTRED(A∪{x}) and x∉Ext, then Ext∈ EXTRED(A). In fact, according to x∉Ext , we know that whether (A∪{x}, B) has updated child nodes or a new child node appears, there must be some Ext∩(A–Ai) not null set, namely, to (A,B) Ext satisfy theorem 1, which will exist nExt∈EXTRED(A) satisfying nExt⊆Ext. By 1) conclusions, we know the nExt∈EXTRED(A∪{x}), which contradicted with Ext having minimum, thus there will be Ext∈ EXTRED(A). The conclusion is established.

2) ①considering the child nodes of (A∪{x},B) only having updated nodes, the extension of some child nodes's changes from Ai to Ai∪{x}, according to the necessary condition of theorem 1, we know any elements of EXTRED(A), namely Ext, meet the conditions to (A∪{x}). It need to state that Ext is minimum below, in fact, if sExt for (A∪{x}) also satisfied theorem 1 and we are easy to know that sExt∈EXTRED (A), which contradicted with Ext∈EXTRED(A), namely Ext, having minimum. Thus, the conclusion is established.

②if (A∪{x}, B) increases the new node (Ci∪{x}, Di)1≤i≤s≤k as its child node, then, for each new node (Ci∪{x},Di), their corresponding extension of the generated value is Ci, and there is a certain A j(1≤ j≤ k) satisfying Ci⊆Aj, thus to any Ext ∈EXTRED(A), Ext to (A ∪ {x}) satisfy theorem 1, and Ext must be minimal value, and this situation is similar to the situation that (A ∪ {x}, B) have updated nodes as child nodes. And we could prove it, thus the conclusion formed.

③if the child nodes of (A∪{x}, B) have both updated nodes and new node, the situation can be proved by multistep computing epitaxial reduction, when (A∪ {x},B) adds a new node or update a child node, according to the ① and ② analysis of the two situations, we will know this EXTRED (A∪{x}) must contain EXTRED (A), so the conclusion is established.

Theorem 2 shows that as the object inserted, the epitaxial reduction function of concept lattice nodes is monotone increasing. Next, discuss the changes of the epitaxial reduction function of each node in the concept lattice, when inserted the object x.

Theorem 3: assume that (A, B) is updated node, {(Ai, Bi)}1≤i≤k are k child nodes of (A,B), if (Ai∪{x}) was the extension of the updated nodes or the new nodes. Note the epitaxial reduction of (A∪{x}, B) is EXTRED1 (A∪{x}) before updating, and the epitaxial reduction is EXTRED2 (A∪{x}) after updating.

Then, EXTRED2 (A∪{x}) = EXTRED1(A∪{x}).

It is proved: at this time, because the necessary condition of the epitaxial reduction of (A∪{x}) does not change, obviously, the conclusion is founded.

Theorem 4: assume that (A, B) is updated node, {(Ai, Bi)}1≤i≤k are k child nodes of (A,B), if (Cj∪{x}) was the extension of the new nodes. Note the epitaxial reduction of (A∪{x}, B) is EXTRED1 (A∪{x}) before updating, and the epitaxial reduction is EXTRED2(A∪{x})

after updating. Then,

EXTRED2(A∪{x})=EXTRED(A)∪MIN{{X∪{y}| ∈EXTRED1(A∪{x}) - EXTRED(A), y∈A–Cj}∪ EXTRED(A)} (1)

It is proved: the necessary condition of the epitaxial reduction of (A∪{x}) increase the condition of (A - Cj) being not empty, attention that at the moment , there is Aj satisfying Cj⊂Aj , if Ext∈EXTRED2(A∪{x}), then the Ext∩(A∪{x}–Aj) is not null set, because we do not take the situation of Ext not containing x into account, so it can be divided into two cases: {Ext–{x}}∩(A–Aj) is not null, at this point it is obviously that Ext∈EXTRED1 (A∪{x}), namely formula (1) set up; {Ext–{x}}∩(A–Aj)=Φ, at this time it will exist sExt∈EXTRED1(A ∪{x}) and sExt⊆Ext, and because Ext∩(A–Cj) is not null, so there will be y∈A–Cj makes sExt∪{y} ⊆Ext. According to Ext have the quality of minimality in EXTRED2 (A∪{x}), we can get sExt∪ {y}=Ext , thereby formula (1) is established. The theorem is proved.

Theorem 2-4 shows the condition of the epitaxial reduction of updated nodes in the process of incremental updating. Then new nodes should be taken into consideration, because the child nodes of the new nodes can only be corresponding with the generated value and the new node, So it should be relatively simple compared to updated nodes. It is similar to the above proof, so here, the relevant conclusions are given without proving.

Theorem 5: (A∪{x}) is the new node extension, A is the generated value extension, and have no other child nodes.

Then EXTRED (A∪{x}) = {t∪{x}|t∈ EXTRED(A)}.

Theorem 6: (A ∪ {x}) is the new node extension, A is the generated value extension, is the generated value extension, {(Ai)}1≤i≤k is the extension of other child node (all of them are new nodes) , note the present epitaxial reduction is EXTRED1 (A∪{x}), if Ak+1 is the extension of new child nodes, note the present epitaxial reduction is EXTRED2(A∪{x}), then, EXTRED2(A∪{x})=MIN{X∪{y}|X∈EXTRED1(A∪{x}), y∈A–Ak+1}. (2)

According to the above argument, we spread the epitaxial reduction calculation of a node to synchronization calculation when inserting a new object every time, on the one hand, it could divided the complex calculation originally into relatively simple step to complete the calculation, on the other hand, the number of nodes that need to recalculate every time are controlled by the connotation of the new object sets, and does not need to other nodes.

In conclusion, we have completely discussed the epitaxial reduction principle of the incremental calculation in the process of incremental constructing concept lattice. Then, based on the above theorem, By modifying the Godin algorithm, we can give the synchronous calculation method of the epitaxial reduction.

4. Iredic Algorithm

according to the above analysis, we know that it should be considered from the following two types of nodes that modifying incremental concept lattice construction algorithm to complete the calculation of epitaxial reduction: updated nodes: after updating, the extension increases x, it should be calculated reduction again according to the theorem 2-4, when its child nodes changed; new nodes: it should be recalculated the reduction again, and calculating its reduction (according to the theorem 2-6) when it adds its corresponding generated value to child nodes.

The synchronous calculation algorithm based on the epitaxial reduction of incremental construction lattice IREDIC.

Input: structure array Lattice and object set G / / Lattice nodes increase the domain ERed of storage reduction

Output: array structure Lattice

Step1: all elements in the G is been built with the incremental method. Note x is an element for G (waiting for inserting), Lattice is the concept Lattice before inserted the x. sorting the nodes in the Lattice from small to large by the connotation potential.

Step2: initializing array Cmax (the array is used to determine generated value or updated nodes. The first dimension is taking the connotation potential for the subscript. The second dimension is based on the Hash technology). Each node in the Lattice oCon:

Step3: spread the Lattice nodes: note oCon for access nodes.

Step4: Rint(oCon) = Intent (oCon)∩ f(x). If the Cmax[|RInt(oCon)|] [RInt(oCon)] is null, please deposit oCon into Cmax[|RInt(oCon)|] [RInt(oCon)], at the moment, the oCon is generated value or updated nodes (Cmax kept in concept, the connotation of preservative concept in Cmax and the intersection of f (x) must be appeared the first time, which is also the the necessary and sufficient conditions of this concept being generated value or updated nodes); Otherwise, turn to step 3 (don't need to deal with the old nodes).

Step5: if the connotation of oCon is contained by f (x), namely, it is updated nodes, update this node and update its parent nodes pmCon, its domain ERed is as follows:

pmCon. ERed=pmCon. ERed∪COMPUTE(x, pmCon, oCon).

Otherwise, oCon is generated value, which generates new nodes nCon and updates edges in turn. And it adds the edges of the new nodes, at the same time, it updates the ERed domain for the corresponding parent node of nCon, which as follows:

nCon.Ered = {t∪{x}|t∈oCon.ERed},

pmCon.ERed=pmCon.ERed∪COMPUTE(x, pmCon, oCon).

Step6: if Lattice does not have the access nodes, make the oCon for the next access nodes, turn Step4.

Step7: if there is not inserted element in G, make x for the next waiting inserted elements, turn to step 3; Otherwise, the program will terminate.

In the algorithm, Step1-Step7 is the outer loop, which used to build incremental concept lattice for all the elements in G; Step 3-Step6 is the inner loop, which used to spread all the nodes in the Lattice and look for generated value and updated nodes. In the Step5, The process of handling Ered domain to updated nodes and new nodes is added respectively based on the Godin algorithm. Among them, the COMPUTE function can be designed as follows according to the theorem 2-6:

COMPUTE (x: a new object, Con: waiting updated Ered domain node, cCon: the Con child nodes)

```
BEGIN
tmp = Con.ERed
A=Extent(Con)–Extent(cCon)
FOR each y in A DO
FOR each X in Con.ERed DO
tmp = tmp∪{X∪{y}}
ENDFOR
tmp = MIN(tmp)
ENDFOR
Con. ERed = / TMP/ update the ERed domain of Con nodes
Return tmp
END {COMPUTE}
```

5. The Analysis and Comparison of Experimental Results

In the previous section, we give the synchronous calculation algorithm IREDIC of the epitaxial reduction based on inserting incremental build lattice of object, in order to verify the effect of algorithm in computing synchronous calculation of connotation reduction, we also need to interchange object sets and connotation sets of the test background and apply the above algorithm to complete the synchronous calculation of the original background based on the connotation reduction of inserting attribute of incremental build lattice (Jiao, J.M. et al, 2013). We use c + + to implement the above two algorithms and the progressive algorithm XRED mentioned in the article of the connotation reduction set calculation(He, C. et al, 2011), in P41.8 G, 256 m, Windows 2000 machine, we use the random background to do the following experiments. In the experiment, the specified object number is 200, attribute number is 30, relationship paternity is 30%, Add object is divided into 5 groups, each group has 100. Figure 1 is the running time of ICIRET and XRED algorithm, the points on the discount show the time required of connotation reduction of concept lattice calculation after completing the insertion of each object set.

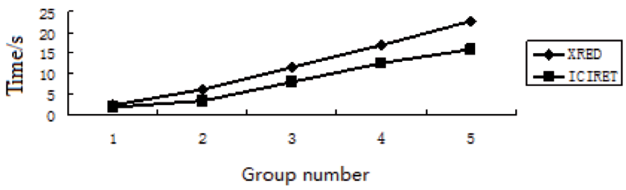

Figure 1. Comparison results of connotation reduction

From the figure 1, we can see that when the connotation reduction in the concept lattice nodes was calculated, the running time of algorithm ICIRET is below XRED algorithm. The main reason that such circumstance appears is that the cost of the calculative minimal value is greater and greater with the parent nodes increase, when XRED calculate the reduction of concept lattice or pruning lattice. However, ICIRET is calculated based on incremental updating, each time, some nodes only need to update the connotation reduction, and each time the connotation reduction of generated new nodes is calculated simply, which reduce the computational complexity of connotation reduction, it also shows that the proposed computing method of synchronous connotation reduction is effective.

6. Conclusion

In this paper, the synchronous calculation of nodes connotation reduction is implemented based on incremental build lattice, the idea of combining the progressive connotation reduction calculation mentioned in literature with incremental structured concept lattice is supplemented and improved (Yang, L. et al, 2012).

Because in the process of updating, the changed nodes of connotation reduction is only updated nodes and new nodes, which reduces the computational cost. In addition, the paper proves the related conclusion of reduction changed in the process of updating, these conclusions provide the theoretical basis for the future design of similar algorithms.

Acknowledgements

This work is supported by: (1) The National Fund Program of Social Science: Research on the “Ecological Co-management” Mechanism between the Interest Bodies in the Process of the New-style Urbanization of Central Plain Economic Region (Grant No. 13CY036); (2) The Philosophy and Social Science Program of Anhui Province: The Empirical Research on the Growth Theory of Green TFP (total factor productivity) and the Economic Growth in Northern Region of Anhui Province (Grant No. AHSK11-12D105).

References

1. Alaoui, H. and Godin, R. Incremental concept formation algorithms -based on Galois (concept) lattices, Journal of Computational Intelligence, 2009, vol. 11, pp: 246-267.
2. Ganter, B. and Wille R. Formal concept analysis: mathematical foundation, New York: Springe -Verlag, l999, pp:236-248.
3. Godin, R. and Missaoui, R. An incremental concept formation approach for learning from the database, The Theoretical Computer Science, 1999, vol. 133, pp: 387-419.
4. Godin, R. and Missaoui,R. An incremental concept formation approach for learning from databases, Theoretical Computer Science, Special Issue on Formal Methods in Databases and Software Engineering, 1999, vol. 133, pp: 3467-489.
5. He, C. and Cheng, X.Q. Mining Hierarchical Concept Lattice for Faceted Navigation, Chinese Journal of Computers, 2011, pp: 1589-1602.
6. Liu, B. and Hsu, W. Integrating classification and association rule mining, New York: Proceedings of the Second International Conference on Knowledge Discovery and Data mining, 1998, pp: 80-86.
7. Liu, W.M. and Yang, J. Research of algorithm based on uncertain data for constrained fre-

quent sets, Application Research of Computers, 2012, iss. 3, pp: 3669-3681.
8. Liu, Z.Q. and Fu, Y.C. Blog community detection based on formal concept analysis, Journal of Computer Applications, 2013, iss. 10, pp: 189-191.
9. Xie, Z.P.and Liu, Z.T. The connotation reduction and calculation of the concept lattice nodes, Computer engineering, 2001, vol. 27, PP: 9-11.
10. Yang, L. and Wang, Y.H. Linguistic truth-valued concept lattice based on graded linguistic values chain and its application, Journal of Computer Applications, 2012, iss. 9, pp:2523-2526.

The Passenger Demand Prediction for Airport Line of Rail Traffic

Luo Xiao-qiang and Wang Lei

School of Architecture, Chang' an University, Xi'an, China

Corresponding author is Luo Xiao-qiang

Abstract
The passenger flow prediction use the construction necessity and feasibility of the rail traffic of airport line as an important base. Based on the analysis of characteristics of passenger flow, it states the trip distribution prediction based on construct matrix method and predication method and corresponding model which are split based on SP questionnaire survey. It shows that the model has a direct and practical characteristic from an example.
Keywords: AIRPORT, RAIL TRAFFIC, PREDICTION AND ANALYSIS OF PASSENGER FLOW, CONSTRUCT MATRIX METHOD, DISAGGREGATE MODEL

1. Introduction

With the rapid development of China social and economy in recent years, there is an increased demand for high-speed passengers and cargos transportation among domestic and international. The transportation volume of air grow rapidly, which lead to ground passenger transportation system base on original roads could not keep pace with the requirement of air transportation in speed, convenient and comfortable. Rail is getting deeply attention and wide application in urban-airport connection system for the feature in speed and huge-volume.

There are many research conclusions in rail transportation witch connected airport. The relatively important results of them are: Rong-Chang Jou et. al. (2011) investigated the airport ground access mode choice behaviours of air passengers who are travelling overseas from Taiwan, and fund out that while out-