Mining FCI of Incremental Attribute in Iceberg Concept Lattice

Anrong Xue  
School of Computer and Telecommunication  
Jiangsu University  
Zhenjiang, China  
xuear@ujs.edu.cn

Fuqiang Wang  
School of Computer and Telecommunication  
Jiangsu University  
Zhenjiang, China  
wfq_2012@sina.com

Mingcai Zhang  
School of Computer and Telecommunication  
Jiangsu University  
Zhenjiang, China  
zhzhmc86@163.com

Ming Li  
School of Computer and Telecommunication  
Jiangsu University  
Zhenjiang, China  
fa2050@126.com

Abstract—Frequent closed itemsets is a smaller, representative of the frequent itemsets. Some existed algorithms for mining frequent closed itemsets based on FP-Tree improvement or incremental mining the database with objects, thus, mining frequent closed itemsets inefficient in the database of increasing attribute. This paper, based on Iceberg concept lattice model, presents a frequent closed items sets mining algorithm FCI-AI in the Iceberg concept lattice. Adding an attribute to database, our algorithm can use the original Iceberg concept lattice for mining the frequent closed itemsets incrementally without having to mine all frequent closed itemsets from scratch. The experimental results show that proposed FCI-AI algorithm with the advantage of less repetitive tasks and efficiency.

Keywords- Frequent Closet Itemsets; Iceberg Concept Lattice; Sub-Context; Sub-Lattice

I. INTRODUCTION

The solution of frequent itemsets in association rule mining\[1\] is the basis and precondition and also is the most time-consuming step. Reducing the number of candidate itemsets is the best method of reducing costs. Frequent closed itemsets (FCI) is implied all frequent itemsets, so we can convert the problem of frequent itemsets to the problem of frequent closed itemsets. Mining frequent closed itemsets can effectively reduce the scale of itemsets, which will make it meaningful for users to find the knowledge of dense database.

Currently, there are frequent closed itemsets mining algorithms as follows: Han J proposed the depth-first algorithm for mining FCI, and exploit a high compressed FP-Tree structure to store the database. After this, they present Closet+ Algorithm and Grahne proposed FP-Close algorithm\[6\]. All of these algorithms are based on the optimization of FP-tree. Their common drawback is that if the tree is a dense generated FP (the worst case is a full prefix tree), the performance of algorithms significantly decreased, the efficient of FCI is low and it will make much repeated work for mining dynamic database. Godin's GALICIA-T algorithm\[4\] based on the concept lattice was the dynamic database incremental algorithm for mining frequent closed itemsets, they use concept lattice for mining dynamic datasets of FCI, But they are applied only to dynamic database with increasing objects. In fact, changes in the database should be including two aspects: First, mutative object in database, and second, changes in attribute. To solve the problem of mining FCI in the database of attribute increasing, based on Iceberg concept lattice model, this paper proposes incremental algorithm FCI-AI (Attribute Incremental Algorithm for Frequent closet Itemsets) for mining frequent closet itemsets in dynamic database. The algorithm solves the problem of requiring to re-excavate existed FCI and basing on the original FCI production of the new issues, and has better scalability.

II. RELATED CONCEPTS

A. Frequent Closed Itemsets

Let I = {i1, i2, ..., id} be the attribute set in a given database D. An itemsets X is a subset of I, A transaction T of D is also a subset of I, which is associated with a unique transaction identifier TID. The support of X is the percentage of transactions in D, denoted by sup(X). The association rule is the implication of the form X→ Y, where X⊂I,Y⊂I and X∩Y=Φ. If sup(X) is not less than the user defined minimum support threshold min-sup, X is called a frequent itemsets in D.

Definition 1 If an item set X satisfy the following two conditions:1).X is frequent itemset;2).There is no direct superset Y of X, make sup(X)=sup(Y),we called itemsets X is a frequent itemsets, sup(X) is the support threshold of X.

As the frequent information of all the frequent itemsets can be made uniquely by the frequent closed itemsets and the computational cost of FCI is much less than the frequent itemsets. Thus, it is an effective way to mine the frequent closed itemsets in association rule.

B. Iceberg Concept Lattice

Concept lattice is the core structure of formal concept analysis. Each node in Concept lattice is a concept which consists of two parts: Extent, the object set covered by concept, denoted by Extent(concept); Intent, the common features of Extent(concept), denoted by Intent(concept).

In formal concept analysis, a formal context is a triple T = (O, A, R), where O = {o1, o2, ..., on} is a set of objects, A = {a1, a2, ..., am} is a set of attributes and the binary relation R(o, a) means that the object o has the attribute a.

Table 1 represents a formal context of concept lattice. In the table, {1,2,3,4,5} and {a,b,c,d} are object set and attribute set of the formal context respectively. X express that an object has an attribute.
A concept $C_1=(X_1,Y_1)$ is a subconcept of a concept $C_2=(X_2,Y_2)$ if $X_1 \subseteq X_2$ (or dually $Y_2 \subseteq Y_1$) and we write $X_1 < C_2$ and there is no other element $C_3$ in the lattice such that $C_1 < C_2 < C_3$. $C_1$ is called parent of $C_2$ and $C_2$ child of $C_1$.

Definition 2 A concept $C_3$ is called frequent concept if its extent is frequent, its support denoted by $sup(C)$, $sup(C) = \left| Extent(C) \right| / \left| O \right|$ are the count of all objects.

Definition 3 The set of all frequent concepts of a formal context $T$ together with the partial order relation $\leq$ forms a so-called Iceberg concept lattice $IL(T)$.

![Figure 1. Iceberg concept lattice](image)

Note that an Iceberg lattice is an order filter of the complete concept lattice and in general only a join semi-lattice. The existing algorithms of building Iceberg concept lattice: Titanic algorithm, snow, and new Border algorithm. For instance, the Iceberg concept lattice of the formal context of Table 1 by min-sup = 30% is shown in Figure 1.

III. FCI-AI ALGORITHM

A. Thought of the algorithm

In certain support conditions, each frequent concept in Iceberg concept lattice corresponds with a frequent closed itemsets. That is all intents of frequent concepts are FCI. Therefore, the process of mining FCI can be reduced as building Iceberg concept lattice incrementally with attribute increasing, and then output the intents of every frequent concept. Set the original formal context of $T = (O, A, R)$ corresponding to the original Iceberg concept lattice $IL(T)$ and new attribute $a^*$, its object set is $g(a^*)$ and $g(a^*) \subseteq O$. Incremental construction Iceberg concept lattice is generated in the base of original Iceberg concept lattice $IL(T)$ and new attribute $a^*$, to obtain the Iceberg concept lattice $IL(T^*)$ corresponding to the formal context $T^* = (O, A \cup \{a^*\}, R^*)$.

According to the relationship between $g(a^*)$ and the original Iceberg concept lattice, it can be defined the frequent new concept, frequent updated concept and the frequent constant concept. Their definitions as following.

Definition 4 Frequent new concept: When a new attribute $a^*$ insert into formal context $T$, generated a concept $C$ and its support $sup(C) \geq min-sup$. There do not exist one concept that its extent is the same with Extent($C$) in original $IL(T)$, we can say that $C$ is a frequent new concept.

Definition 5 Frequent updated concept: Let $C_1$ is a frequent concept in original $IL(T)$, if $Extent(C_1) \subseteq g(a^*)$, $C_1$ is called a frequent updated concept. The extent and intent $C_1$ would be updated as $\{Extent(C_1), Intent(C_1) \cup \{a^*\}\}$.

Definition 6 Frequent constant concept: The frequent concept, nothing to do with the new attribute $a^*$ and $g(a^*)$, retain to new generation of Iceberg concept lattice $IL(T^*)$ from the original Iceberg concept lattice $IL(T)$.

As the relationship between $g(a^*)$ and the extent of frequent concept in original $IL(T)$, our algorithm FCI-AI can be divided into two cases to discuss.

Firstly, we introduce a property about frequent updated concept before conducting the first case.

Property 1 Let a frequent concept $C=\{Extent(C), Intent(C)\}$ and attribute $a^*$, its object set is $g(a^*)$, there is a relation $Extent(C) \subseteq g(a^*)$. Insert $a^*$ into the extent of $C$, $C=\{Extent(C), Intent(C) \cup \{a^*\}\}$. Thus, $C$ is still a frequent concept.

Proof: The common object sets of $\{Extent(C), \cup \{a^*\}\}$ are $\{Extent(C) \cup \{a^*\}\}$. Because of $\{Extent(C), \cup \{a^*\}\} \subseteq g(a^*)$, $\{Extent(C) \cup \{a^*\}\} \geq min-sup$. Thus, $C$ is updated as $C=\{Extent(C), Intent(C) \cup \{a^*\}\}$. It is still a frequent concept.

The first case, when a new attribute $a^*$ and its objects set are inserted into the original formal context $T$, if there is existing a frequent concept $C$ in the original concept lattice $IL(T)$, we say $C$ is the frequent updated concept. In the case, updating the original Iceberg concept lattice is relatively simple, only to directly update the intents $\{a^*\}$ of frequent concept $C$ and all its subconcepts. By this, we have completed the updated Iceberg concept lattice $IL(T^*)$. Then, modify the frequent closed itemsets corresponding to the frequent updated concept.

The second case, there is not existing a frequent concept $C$ that its extent $\{Extent(C) = g(a^*)\}$ in original Iceberg concept lattice $IL(T)$, The following three problems need to be solved:

a) Generate new concepts, and their support threshold greater than min-sup.

b) Update the frequent updated concept in original $IL(T)$.

c) Update or add the covering relations between frequent concepts.

Firstly, it is needed to extract the Sub-Context form new formal context $T^*$ after an attribute $a^*$ and objects $g(a^*)$ inserted into original formal context $T$.

Definition 7 Let original formal context $T = (O, A, R)$, it is updated as new formal context $T^* = (O, A \cup \{a^*\}, R^*)$ that an attribute $a^*$ and objects $g(a^*)$ inserted into $T$. Extracted a smaller formal context $T_{sub} = (g(a^*), A \cup \{a^*\}, R_{sub})$ from $T^*$, it is known as Sub-Context.

Construction of the Iceberg concept lattice under the Sub-Context is called Sub-Lattice. Because the objects of Sub-Context are smaller than $T$. In order to make the
support of frequent concept in Sub-Lattice relatively unchanged, the support of construction Sub-Lattice should be supcsub:

\[ \text{supcsub} = \frac{\text{supc} \times |O|}{|g(a*)|} \]  

(1)

Then, merge the original \( IL(T) \) and Sub-Lattice, The merging method is as follow:

For each frequent new concept \( C_{\text{new}} \), which were computed their parent concept, denoted by Parent \( (C_{\text{new}}) \) in the original Iceberg concept lattice \( IL(T) \),connect each parent concept to \( C_{\text{new}} \) \( \text{Parent}(C_{\text{new}}) \rightarrow C_{\text{new}} \) if frequent concept \( C \) in \( IL(T) \) and \( C \) in Sub-Lattice have the same extent \( (\text{Extent}(C) = \text{Extent}(C_1)) \); therefore, \( C \) is a frequent updated concept, only to directly update the intents \( \cup[a^*] \) of frequent concept \( C \) and all its subconcepts, connect the parent concept of \( C_1 \) in Sub-Lattice \( (\text{Parent}(C_1) \rightarrow C_1) \); then, delete \( C_1 \) and its all bottom concept; If \( C \) is a frequent updated concept and \( \text{Parent}(C) \) is root concept in \( IL(T) \), we should delete the connection line with root concept. Lastly, modify the frequent closed itemsets corresponding to the frequent updated concept and add the intent of frequent new concept to FCI set.

B. Algorithm description

According to the thought of the algorithm described above, algorithm FCI-AI proposed to mine FCI in the dataset with its attribute increasing. The pseudo-code is the following:

Input: original formal context: \( T = (O,A,R) \), original Iceberg concept lattice: \( IL(T) \), new attribute \( a^* \) and objects \( g(a^*) \), minimum support threshold \( \text{min-sup} \).

Output: new Iceberg concept lattice \( IL(T^*) \) generated from \( T^* = (O,A \cup \{a^*\},R^*) \) and FCI set.

BEGIN

//Initialize the set of FCI
\( F = \{\} \);

IF exist \( C \subset IL(T) \) and \( \text{Extent}(C) = g(a^*) \) THEN

// \( C \) is a frequent updated concept

\{ 
//delete FCI corresponding with \( C \)
\( F := F \setminus \{\text{Intent}(C),\text{Intent}(\text{Children}(C))\} \);

//update the intents \( \cup[a^*] \) of frequent concept \( C \) and all its subconcepts
\( C_1 = \{\text{Extent}(C),\text{Extent}(C_1) \cup \{a^*\}\}; \)
\( \text{All Intent}(\text{Children}(C)) \cup \{a^*\} \);

//add the intent of \( C \) to \( F \)
\( F_1 := F \cup \{\text{Intent}(C),\text{Intent}(\text{Children}(C))\} \);

\}
ELSE {

//extract the Sub-Context
\( T_{\text{Sub}} := \{g(a^*), A \cup \{a^*\}, R\}; \)

//the support of building Sub-Lattice
\( \text{supc}_{\text{Sub}} = \text{supc} \times |O| / |g(a^*)| \);

Build Sub-Lattice \( SIL(T^*) \) from Sub-Context

Merge \( SIL(T_{\text{Sub}}) \) and \( IL(T) \) to \( IL(T^*) \);

\( F := F \cup \{\text{FCI} \text{ corresponding with frequent updated concept in } IL(T)\} \);

Find Parents in \( IL(T) \) for \( \forall C_1 \in SIL(T_{\text{Sub}}) \);

// add all intents of all the frequent concept in \( SIL(T_{\text{Sub}}) \) into \( F \)
\( F := F \cup \{\text{all Intent in } SIL(T_{\text{Sub}})\} \);

\}

END

C. Example

In this section, we will take an example to illustrate our algorithm FCI-AI. In the condition of minimum support threshold \( \text{min-sup} = 30\% \), each intent of frequent concept in original \( IL(T) \) which build from formal context \( T \) in Table 1 is a frequent closed itemsets. So the set of FCI in original formal context is \( \{a,b,c,d,ac,ad,bd\} \). we insert an attribute \( e \) and its objects \( g(e) = \{1,3,5\} \) into Table 1. Owing to existing a frequent concept \( \{1,3,5\} \) in \( IL(T) \), its extent is identical to \( g(e) \). So \( \{1,3,5\} \) is a frequent updated concept. In order to incrementally generate new Iceberg concept lattice \( IL(T^*) \) based on the original \( IL(T) \), we barely need to modify the intent \( \cup[e]\) of \( \{1,3,5\} \) and its subconcepts \( \{1,3\}, \{a,d\}, \{3,5\}, \{b,d\} \).

In the figure 2, it is the updated \( IL(T^*) \) based on Figure 1. Then, we should update the FCI which is \( \{d,ad,bd\} \) corresponding with frequent updated concept in \( IL(T) \). Finally, the set of FCI is \( \{a,b,c,de, ac, ade, bde\} \).

![Figure 2. updated Iceberg concept lattice](image)

The second case, we insert an attribute \( f \) and its objects \( g(f) = \{1,4,5\} \) into Table 1. Because of do not exist a frequent concept which its extent is identical to \( g(f) \) in original \( IL(T) \), we should extract Sub-Context at first. In Table 2, it is a Sub-Context which include objects set \( g(f) \) and all attribute \( \{a,b,c,d,f\} \). Sub-Lattice is constructed by Sub-Context, illustrating in Figure 3. The support of Sub-Lattice is \( \text{supc}_{\text{Sub}} = 0.3*5/3 = 0.5 \).

![Figure 3. Sub-Lattice](image)
Then, we merge the original $IL(T)$ in Figure 1 and Sub-Lattice in Figure 3. In the Sub-Lattice, concept $((1,4,5),\{f\})$ and $((1,5),\{d,f\})$ are frequent new concept, finding their parent concepts $((1,2,3,4,5),\{\})$ and $((1,3,5),\{d\})$ respectively in original $IL(T)$. As in Figure 1, there is extent of frequent concept $((1,4),\{a,c\})$ is identical to $((1,4),\{a,c,f\})$ in Figure 3, so $((1,4),\{a,c\})$ is a frequently updated concept. Figure 4 illustrate the merged Iceberg concept Lattice.

Lastly, we update the set of FCI which correspond with frequent updated concept $((1,4),\{a,c\})$ as $\{acf\}$ and add the intents of frequent new concept $((1,4,5),\{f\})$ and $((1,5),\{d,f\})$ into FCI set. Finally, the set of FCI is $\{a,b,c,d,f,ad,bd,df,acf\}$.

IV. EXPERIMENTAL AND ANALYSIS

In order to verify the effectiveness of the proposed FCI-AI based on Iceberg concept lattice for mining frequent closed itemsets with attribute increasing. Our experiments are implemented on the same platform and machines: Intel (R) Core(TM)2 Duo CPU T5800 2.1G , 1G Windows XP Professional, and JAVA Programming Language, and compared to the previous algorithm GALICIA-T. Experimental test data are mushrooms provided by UCI data sets, the datasets have 8124 records, describes 23 properties of mushrooms, each of which has enumerated values 2 to 12. We consider the part of the datasets as the original formal context of the algorithm; the interception of this section contains all the objects and the first 5 properties, and then inserted the rest of the property to the original formal context one by one. Based on the original concept lattice, verify the efficiency of incremental mining.

In the conditions of different support, the experiment compares the consuming time of mining out all the FCI of two algorithms by inserting a new attribute into the original formal context. As figure 5 shown, horizontal axis indicated the number of attributes, the vertical axis, said the consuming time of mining out all the FCI in certain conditions after inserting the attributes into the original formal context one by one. As figure 6 shown, select a different support (20%, 15%, 10%, 5%, 1%) to compare performance of two algorithms.

From the figures above, the running time of GALICIA-T algorithm is more than FCI-AI in the same formal context on the FCI mining. Firstly, the reason is that it is incremental mining algorithm based on the frequent closed itemsets of concept lattice which needs to construct complete lattice, when the support is relatively large, the number of non-frequent concepts are much more than the number of frequent concepts, thus FCI-AI algorithm needs only to calculate the frequent concept to build Iceberg concept lattice, so this will reduce the production of non-frequent concepts and improve the efficiency of the algorithm; Secondly, when add a new property to the original form of the background, GALICIA-T will re-construct concept lattice to calculate FCI, which will generate a lot of repeated FCI, thus the FCI-AI algorithms can take advantage of the original concept lattice, and on the basis of the original excavation FCI , only update the FCI which is generated after adding new properties. Therefore, FCI-AI algorithm has the advantages of less duplication of work and high efficiency.

V. CONCLUSIONS

In this paper, we propose mining FCI with attribute increasing in Iceberg concept lattice. After adding a property into the original formal context, we do not need to re-excavation of all the FCI, and only calculate the updated new FCI on the basis of the original Iceberg concept lattice, so the algorithm has better reusability and higher efficiency in the implementation. The disadvantage of FCI-AI is that the more objects of a new attribute, the less performance of the algorithm. Further work is mainly to combine Iceberg concept lattice with frequent closed itemsets and enhance the efficiency of mining FCI in the Iceberg concept lattice.
ACKNOWLEDGMENT

This work is sponsored by National Natural Science Foundation of China under Grant No.60773049 and senior personnel launched Fund of Jiangsu University No.09JDG041.

REFERENCES


