Fast Construction of Concept Lattice

Naveen Kumar, Anamika Gupta and Vasudha Bhatnagar

Department of Computer Science, University of Delhi, India
nk,agupta,vbhatnagar(@cs.du.ac.in)

Abstract. The proposed algorithm computes concepts and lattice structure together in one step avoiding repeated computation of same concepts. The time complexity is $O((|G| + |M|)|M|)$ per lattice element where G is set of objects and M is set of attributes.

1 Introduction

Concept Lattice is a core structure in Formal Concept Analysis. In this paper, we propose an algorithm that generates the lattice structure along with the concepts in a single step. The worst case complexity of proposed algorithm is $O(|G| + |M|)|M|)$ per lattice element which is better than worst case complexity of algorithms proposed by Bordat, Ganter-Alaoui and Chein-Alaoui as mentioned in [1] as well as the algorithm proposed by Nourine [3].

2 Approach for Computing the Concept Lattice

Observation 1 *For a pair of concepts* (X_1, Y_1) *and* (X_2, Y_2) *computed at level k in the algorithm,* $X_1 \cap X_2$ *will exist at level k+1 if and only if* (X_1, Y_1) *and* (X_2, Y_2) *are the sibling concepts. Level k concepts have k attributes in the intent.*

Observation 2 *Let (X,Y) be a concept where* $|Y| = m$, $|M|$ *is total number of attributes, (X,Y) can have at most* $(|M| - m)$ *children, each parent of (X,Y) can have at most* $(|M| - (m-1))$ *child concepts. This implies that (X,Y) can have at most (|M|-m) sibling concepts per parent. In the worst case, all possible concepts are present, (X,Y) can have at most m parents.*

Proposition 1. *Let (X,Y) be a concept with parent set P(X,Y) =* $\{P_1, P_2, \ldots P_n\}$, $|P| = n$. *Let* S_i *be the set of child concepts of* P_i *for all* $i = 1, .., n$. *To compute C(X,Y), the set of child concepts of (X,Y), it is sufficient to compute the intersection of X with extent of each concept* $\epsilon\ S_i$ *for exactly one i,* $i\ \epsilon\ 1..n$.

The algorithm computes the concepts based on above proposition, thus avoids repeated computation of the same concepts.

2.1 Algorithm

Input: Context: G x M where G, M are objects and attributes respectively
Output: Concept Lattice L
01. Generate the root concept (G, ϕ) in L
02. for each attribute A_i, find $A_i^{'}$
03. if for some $j < i$, $A_i^{'} = A_j^{'}$
04. add the edge $(A_i^{'}, (A_i \cup A_j)) \rightarrow$ root to L
05. remove the edge $(A_i^{'}, A_i) \rightarrow$ root from L
06. else add the edge $(A_i^{'}, A_i) \rightarrow$ root to L
07. for each pair (X_1, Y_1) in the lattice L
08. for each sibling pair (X_2, Y_2) of a parent (P_1X_1, P_1Y_1) of (X_1, Y_1)
09. find intersection of X_1 with extent of its sibling pair (X_2, Y_2)
10. $X = X_1 \cap X_2$, $Y = Y_1 \cup Y_2$
11. if $X \neq X_1$ and $X \neq X_2$
12. if X does not belong to extent list l // stored as trie
13. add (X, Y) to concept lattice L, add X to extent List l
14. add edge $(X, Y) \rightarrow (X_1, Y_1)$, $(X, Y) \rightarrow (X_2, Y_2)$ to L
15. for all other parents (P_2X_1, P_2Y_1) of (X_1, Y_1)
16. add edge $(X, Y) \rightarrow (X_3, Y_3)$ to L such that
17. $Y_3 = (P_2Y_1 \cup m)$ where $(P_1Y_1 \cup m) = Y_2$, $m \epsilon M$
18. else for some (X_3, Y_3) in the extent list l where $X = X_3$
19. add edge $(X_3, Y_3) \rightarrow (X_1, Y_1)$ $(X_3, Y_3) \rightarrow (X_2, Y_2)$ to L
20. if $P_1X_3 = P_1X_1$ or $P_1X_3 = P_1X_2$
21. remove the edge $(X_3, Y_3) \rightarrow P_1X_3, P_1Y_3)$ from L
22. else if $X = X_1$ // i.e. (X_1, Y_1) is child of (X_2, Y_2)
23. add an edge $(X_1, Y_1) \rightarrow (X_2, Y_2)$ to L
24. remove the edge $(X_1, Y_1) \rightarrow (P_1X_1, P_1Y_1)$ from L
25. else if $X = X_2$ // i.e. (X_2, Y_2) is child of (X_1, Y_1)
26. add an edge $(X_2, Y_2) \rightarrow (X_1, Y_1)$ to L
27. remove the edge $(X_2, Y_2) \rightarrow (P_1X_1, P_1Y_1)$ from L

3 Experimentation and Results

Experiments on both sparse and dense data sets reveal improvements over Nourine's[3] and Bordat's algorithm[1]. Results of Nourine's[3] and Bordat's algorithm[1] are obtained using "lattice Generator" developed by Kuznetsov and Obiedkov[2].

References

1. Godin R., Missaoui R., Alaoui H.: Incremental Concept Formation Algorithms Based on Galois Lattices, Computation Intelligence (1995)
2. Kuznetsov S.O.: Ob"edkov S.A.: Comparing performance of algorithms for generating Concept Lattices, JETAI (2002)
3. Nourine L. Raynaud O.: A Fast Algorithm for Building Lattices, Information Processing Letters, vol. **71** (1999) 199–204