An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data

Akihiro Inokuchi*, Takashi Washio, and Hiroshi Motoda

I.S.I.R., Osaka University
8-1, Mihogaoka, Ibarakishi, Osaka, 567-0047, Japan
Phone: +81-6-6879-8541 Fax: +81-6-6879-8544
washio@sanken.osaka-u.ac.jp

Abstract. This paper proposes a novel approach named AGM to efficiently mine the association rules among the frequently appearing substructures in a given graph data set. A graph transaction is represented by an adjacency matrix, and the frequent patterns appearing in the matrices are mined through the extended algorithm of the basket analysis. Its performance has been evaluated for the artificial simulation data and the carcinogenesis data of Oxford University and NTP. Its high efficiency has been confirmed for the size of a real-world problem. ...

1 Introduction

Mining knowledge from structured data is a major research topic in recent data mining study. “Graph structure” is one of the representatives of the structured data since it frequently appears in real-world data such as web links and chemical compounds structures. In the field of chemistry, CASE and MultiCASE systems have been often used to discover characteristic substructures of chemical compounds [8], [9]. Though these systems can efficiently find the substructures, the class of the substructures is limited to the no-branching atom sequences. Wang and Liu proposed the mining of wider class of substructures which are subtrees called schemas [14]. Though the proposed algorithm is very efficient to mine frequent schemas from massive data, the mining patterns are still limited to acyclic graphs. To mine characteristic patterns having general graph structures, the propositional classification techniques, e.g., C4.5, the regression tree techniques, e.g., M5, and the inductive logic programming (ILP) techniques have been applied in the carcinogenesis predictions of chemical compounds [10], [7]. However, these approaches can discover only limited types of characteristic substructures, because the graph structures must be pre-characterized by some specific features and/or ground instances of predicates.

Recently, a technique to mine the frequent substructures characterizing the carcinogenesis of chemical compounds has been proposed without requiring any conversion of substructures to specific features by Dehaspe et al. [3]. They used

[*] Currently beeing in Tokyo Research Institute, IBM, 1623-14 Shimotsuruma, Yamatoshi, Kanagawa, 242-8502, Japan.

D.A. Zighed, J. Komorowski, and J. Żytkow (Eds.): PKDD 2000, LNAI 1910, pp. 13–23, 2000.

the ILP framework combined with levelwise search to minimize the access frequency to the database [11]. Since the efficiency achieved by this approach is much better than the former ILP approaches, some new discovery of substructures characterizing carcinogenesis was expected. However, the full search space was still so large that the search had to be limited within the 6th level where the substructures are represented with 6 predicates at maximum, and they reported that significant substructures have not been obtained within the search level. Some other researches have also developed the techniques to mine the frequent substructures in graph data. The graph-based induction (GBI) is an approach to seek the frequent patterns by iteratively chunking the vertex pairs that frequently appear [12]. SUBDUE is another approach to seek the characteristic graph patterns to efficiently compress the original graph in terms of MDL principle [2]. These approaches do not face the severe computational complexity. However, they may miss some significant patterns, since their search strategies are greedy.

Though the task tackled by these works involves the problem of deciding graph isomorphism which is known to be NP, each work mines some characteristic graph substructures by introducing the limitations on the search space and/or the class of substructures. The objective of this paper is 1) to propose a novel approach named as “Apriori-based Graph Mining”, AGM for short, to mine the frequent substructures and the association rules from the general class of graph structured data in a more efficient manner than the preceding work, and 2) to assess the performance of the approach for the artificially simulated data and also for the carcinogenesis data of Oxford University and National Toxicological Program (NTP) [13].

2 Principle of Mining Graph Substructures

The methods studied in the mathematical graph isomorphism problem are not directly applicable to our case, because the methods are only to check if the two given graphs are isomorphic [4]. We introduce the mathematical graph representation of “adjacency matrix” and to combine it with an efficient levelwise search of the frequent canonical matrix code [5]. The levelwise search is based on the extension of the Apriori algorithm of the basket analysis [1].

2.1 Representation of Graph Structures

A graph in which the vertices and edges have labels is mathematically defined as follows.

Definition 1 (Graph having Labels) *Given a set of vertices* $V(G) = \{v_1, v_2, ..., v_k\}$*, a set of edges connecting some vertex pairs in* $V(G)$*;* $E(G) = \{e_h = (v_i, v_j) | v_i, v_j \in V(G)\}$*, a set of vertex labels* $L(V(G)) = \{lb(v_i) | \forall v_i \in V(G)\}$ *and a set of edge labels* $L(E(G)) = \{lb(e_h) | \forall e_h \in E(G)\}$*, then a graph* G *is represented as*

$$G = (V(G), E(G), L(V(G)), L(E(G))).$$

This graph G is represented by an adjacency matrix X which is a very well known representation in mathematical graph theory [4]. This transformation from G to X does not require much computational effort.

Definition 2 (Adjacency Matrix) *Given a graph $G = V(G), E(G), L(V(G)), L(E(G)))$, the adjacency matrix X has the following (i, j)-element, x_{ij},*

$$x_{ij} = \begin{cases} num(lb) \; ; e_h = (v_i, v_j) \in E(G) \text{ and } lb = lb(e_h) \\ 0 \qquad\quad ; (v_i, v_j) \notin E(G) \end{cases},$$

where $num(lb)$ is an integer arbitrarily assigned to a label value lb. Moreover, a number $num(lb)$ is assigned to the i-th low (i-th column) of the matrix where $v_i \in V(G)$ and $lb = lb(v_i)$.

Definition 3 (Size of a Graph) *The "size" of a graph G is the number of vertices in $V(G)$, i.e., k in Definition 1.*

Definition 4 (Graph Transaction and Graph Data) *A graph $G = (V(G), E(G), L(V(G)), L(E(G)))$ is a transaction, and graph data GD is a set of the transactions, where $GD = \{G_1, G_2, ..., G_n\}$.*

Each element of an adjacency matrix in the standard definition is either '0' or '1', whereas each element in Definition 2 can have the number of an edge label. This extended notion of the adjacency matrix gives a compact representation of a graph having labeled edges, and enables an efficient coding of the graph as shown later.

The representation of the adjacency matrix depends on the assignment of each vertex to the i-th row (i-th column). To reduce the variants of the representations and increase the efficiency of the code matching described later, the vertices are sorted according to the numbers of their labels. The adjacency matrix of a graph whose size is k is noted as X_k, and the graph as $G(X_k)$.

Definition 5 (Vertex-sorted Adjacency Matrix) *The adjacency matrix X_k of the graph $G(X_k)$ is vertex-sorted if*

$$num(lb(v_i)) \leq num(lb(v_{i+1})) \text{ for } i = 1, 2, ..., k-1.$$

In the standard basket analysis, items within an itemset are kept in lexicographic order [1]. This enables an efficient control of the generation of candidate itemsets. However, the vertex-sorted adjacency matrices do not have such lexicographic order. Thus, a coding method of the adjacency matrices need to be introduced.

Definition 6 (Code of Adjacency Matrix) *In case of an undirected graph, the code $code(X_k)$ of a vertex-sorted adjacency matrix X_k;*

$$X_k = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,k} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,k} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{k,1} & x_{k,2} & x_{k,3} & \cdots & x_{k,k} \end{pmatrix},$$

is defined as

$$code(X_k) = x_{1,1}x_{1,2}x_{2,2}x_{1,3}x_{2,3}x_{3,3}x_{1,4}\cdots x_{k-1,k}x_{k,k},$$

where the digits are obtained by scanning the elements along the columns at the upper triangular part of X_k. In case of a directed graph, it is defined as

$$code(X_k) = x_{1,1}x_{1,2}x_{2,1}x_{2,2}x_{1,3}x_{3,1}x_{2,3}x_{3,2}\cdots x_{k-1,k}x_{k,k-1}x_{k,k},$$

where the digits are obtained similarly to the undirected case, but the diagonally symmetric element x_{ji} is added after each x_{ij} when $i \neq j$.

The method proposed in this paper discovers substructures frequently appearing in the graph transaction data GD. The rigorous definition of the substructure is given as follows.

Definition 7 (Induced Subraph) *Given a graph $G = (V(G), E(G), L(V(G)), L(E(G)))$, an induced subgraph of G, $G_s = (V(G_s), E(G_s), L(V(G_s)), L(E(G_s)))$, is a graph satisfying the following conditions.*

$$V(G_s) \subset V(G), E(G_s) \subset E(G),$$

$$\forall u, v \in V(G_s), (u, v) \in E(G_s) \Leftrightarrow (u, v) \in E(G).$$

When G_s is an induced subgraph of G, it is denoted as $G_s \subset G$.

2.2 Algorithm of AGM

Candidate Generation. The two indices which are identical to the definitions of "support" and "confidence" in the basket analysis are introduced.

Definition 8 (Support and Confidence) *Given a graph G_s, the support of G_s is defined as*

$$sup(G_s) = \frac{number\ of\ graph\ transactions\ G\ where\ G_s \subset G \in GD}{total\ number\ of\ graph\ transactions\ G \in GD}.$$

Given two induced subgraphs G_b and G_h, the confidence of the association rule $G_b \Rightarrow G_h$ is defined as

$$conf(G_b \Rightarrow G_h) = \frac{number\ of\ graphs\ G\ where\ G_b \cup G_h \subset G \in GD}{number\ of\ graphs\ G\ where\ G_b \subset G \in GD}.$$

If the value of $sup(G_s)$ is more than a threshold value $minsup$, G_s is called as a "frequent induced subgraph".

Similarly to the Apriori algorithm, the candidate generation of the frequent induced subgraph is made by the levelwise search in terms of the size of the subgraph. Let X_k and Y_k be vertex-sorted adjacency matrices of two frequent induced graphs $G(X_k)$ and $G(Y_k)$ of size k. If both $G(X_k)$ and $G(Y_k)$ have equal elements of the matrices except for the elements of the k-th row and the k-th column, then they are joined to generate Z_{k+1}.

$$X_k = \begin{pmatrix} X_{k-1} & \boldsymbol{x}_1 \\ \boldsymbol{x}_2^T & x_{kk} \end{pmatrix}, Y_k = \begin{pmatrix} X_{k-1} & \boldsymbol{y}_1 \\ \boldsymbol{y}_2^T & y_{kk} \end{pmatrix},$$

$$Z_{k+1} = \begin{pmatrix} X_{k-1} & \boldsymbol{x}_1 & \boldsymbol{y}_1 \\ \boldsymbol{x}_2^T & x_{kk} & z_{k,k+1} \\ \boldsymbol{y}_2^T & z_{k+1,k} & y_{kk} \end{pmatrix} = \left(\begin{array}{c|c} X_k & \begin{matrix} \boldsymbol{y}_1 \\ z_{k,k+1} \end{matrix} \\ \hline \begin{matrix} \boldsymbol{y}_2^T & z_{k+1,k} \end{matrix} & y_{kk} \end{array}\right), \qquad (1)$$

where X_{k-1} is the adjacency matrix representing the graph whose size is $k-1$, $\boldsymbol{x}_i$ and $\boldsymbol{y}_i (i = 1, 2)$ are $(k-1) \times 1$ column vectors. X_k is called the "first matrix" and Y_k the "second matrix". The following relations hold among the vertex-sorted adjacency matrices X_k, Y_k and Z_{k+1}.

$$lb(v_i; v_i \in V(G(X_k)) = lb(v_i; v_i \in V(G(Y_k)) = lb(v_i; v_i \in V(G(Z_{k+1}))),$$

$$\begin{gathered} lb(v_i; v_i \in V(G(X_k)) \leq lb(v_{i+1}; v_{i+1} \in V(G(X_k)), \\ lb(v_k; v_k \in V(G(X_k)) = lb(v_k; v_k \in V(G(Z_{k+1})), \\ lb(v_k; v_k \in V(G(Y_k)) = lb(v_{k+1}; v_{k+1} \in V(G(Z_{k+1})), \\ lb(v_k; v_k \in V(G(X_k)) \leq lb(v_k; v_k \in V(G(Y_k)). \end{gathered} \qquad (2)$$

Here, $i = 1, \cdots, k-1$. $z_{k,k+1}$ and $z_{k+1,k}$ are not determined by X_k and Y_k. Each can take every integer value $num(lb)$ corresponding to each edge label lb or 0 corresponding to the case that no edge exists between v_k and v_{k+1}. In case of an undirected graph, $z_{k,k+1}$ and $z_{k+1,k}$ must have an identical value. This join procedure of X_k and Y_k creates multiple Z_{k+1}s for all possible value pairs of $z_{k,k+1}$ and $z_{k+1,k}$. Note that when the labels of the k-th vertices v_k of $G(X_k)$ and $G(Y_k)$ are the same, exchanging X_k and Y_k (*i.e.*, taking Y_k as the first matrix and X_k as the second matrix), produces redundant adjacent matrices. In order to avoid this redundant generation, the two adjacency matrices are joined only when Eq.(3) is satisfied. The vertex-sorted adjacency matrix generated under this condition is called a "normal form".

$$code(\text{the first matrix}) \leq code(\text{the second matrix}) \qquad (3)$$

In the standard basket analysis, the $(k+1)$-itemset becomes a candidate frequent itemset only when all the k-sub-itemsets are confirmed to be frequent itemsets. Similarly, the graph G of size $k+1$ is a candidate of frequent induced subgraphs only when all adjacency matrices generated by removing from the graph G the i-th vertex v_i $(1 \leq i \leq k+1)$ and all its connected links are confirmed to be frequent induced subgraphs of the size k. As this algorithm generates only adjacency matrices of the normal form in the earlier (smaller) k-levels, if the adjacency matrix of the graph generated by removing the i-th vertex v_i is non-normal form, it must be transformed to a normal form to check if it matches one of the normal form matrices found earlier. An adjacency matrix X_k of a non-normal form is transformed into a normal form X'_k by reconstructing the matrix structure in a bottom up manner. First, an adjacency matrix of the size 1×1 is set for each vertex $v_i \in G(X_k)$. Then, the pair of the matrices for the vertices $v_i, v_j \in G(X_k)$ satisfying the constraints of Eq.(2) and (3) are joined by the operation of Eq.(1). At this time, the values of the elements for (v_i, v_j) and (v_j, v_i) in the original X_k are substituted to the non-diagonal elements $z_{1,2}$ and $z_{2,1}$ respectively to reconstruct the structure of $G(X_k)$. Subsequently, the pair of the obtained 2×2 matrices are further joined according to the constraints of Eq.(1), (2) and (3). The values of the elements $z_{2,3}$ and $z_{3,2}$ are determined from X_k in the similar manner. This procedure is repeated until a $k \times k$ matrix X'_k is obtained. Because X'_k precisely reflects the structure of $G(X_k)$, and is constructed by following the constraints, X'_k is a normal form of X_k. This reconstruction is called "normalization". In the intermediate levels, the normal forms of all induced subgraphs of $G(X_k)$ can be derived. This feature of the normalization is used in the frequency calculation explained latter. The normalization consists of the set of permutations of the rows and the columns of the original matrix X_k. Thus X'_k has the relation of $X'_k = (T_k)^T X_k T_k$ where T_k is the transformation matrix. The details of the normalization can be found in [6].

Canonical From. After all candidate induced subgraphs are derived, the support value of each candidate is counted in the database. However, the normal form representation is in general not unique for a graph. For instance, the following two matrices which are both normal forms represent an identical undirected graph with a unique link label.

$$X_3 = \begin{pmatrix} 0\,0\,1 \\ 0\,0\,1 \\ 1\,1\,0 \end{pmatrix}, Y_3 = \begin{pmatrix} 0\,1\,1 \\ 1\,0\,0 \\ 1\,0\,0 \end{pmatrix}.$$

If the support value is counted for each representation independently, it has to be summed up to obtain the correct support value for the corresponding graph. To perform this summation efficiently, all normal forms for an identical induced subgraph must be indexed. For this purpose, canonical form is defined for normal forms of adjacency matrices representing an identical induced subgraph, and an efficient method to index each normal form to its canonical form is introduced.

Definition 9 (Canonical Form) *Given a set $NF(G)$ of all normal forms of adjacency matrices representing an identical graph G, its canonical form X_c is defined as X having the minimum code number in $NF(G)$, i.e.,*

$$X_c = arg \min_{X \in NF(G)} code(X).$$

We assume that all the transformation matrices S_{k-1} to the canonical form from the normal forms of every frequent induced subgraph of size k-1 are known. Let X^m_{k-1} be the matrix obtained by removing the m-th vertex v_m $(1 \leq m \leq k)$ from $G(X_k)$. X^m_{k-1} is transformed to one of its normal forms, X'^m_{k-1}, by the aforementioned normalization, and thus its transformation matrix T^m_{k-1} is known. Furthermore, let S_{k-1} of X'^m_{k-1} be S^m_{k-1}, then the transformed canonical form is represented by $(T^m_{k-1}S^m_{k-1})^T X^m_{k-1} T^m_{k-1} S^m_{k-1}$. The canonical form X_{ck} of X_k and the matrices S^m_k, T^m_k to transform X_k to X_{ck} are obtained from S^m_{k-1}, T^m_{k-1} by the following expressions. The detailed proof of this transformation can be found in [6].

$$s_{ij} = \begin{cases} s^m_{ij} \; 0 \leq i \leq k-1 \text{ and } 0 \leq j \leq k-1, \\ 1 \quad i = k \text{ and } j = k, \\ 0 \quad \text{otherwise,} \end{cases}$$

$$t_{ij} = \begin{cases} t^m_{ij} \quad\; i < m \text{ and } j \neq k, \\ t^m_{i-1,j} \; i > m \text{ and } j \neq k, \\ 1 \quad\;\; i = m \text{ and } j = k, \\ 0 \quad\;\; \text{otherwise,} \end{cases}$$

$$X_{ck} = arg \min_{m=1,\cdots,k} code((T^m_k S^m_k)^T X_k (T^m_k S^m_k)),$$

where s_{ij}, s^m_{ij}, t_{ij} and t^m_{ij} are the elements of matrix S^m_k, S^m_{k-1}, T^m_k and T^m_{k-1} respectively. $T^m_k S^m_k$ which minimize the code is S_k of X_k.

Frequency Calculation. Frequency of each candidate induced subgraph is counted by scanning the database after generating all the candidates of frequent induced subgraphs and obtaining their canonical forms. Every transaction graph G in the database can be represented by an adjacency matrix X_k, but it may not be a normal form in most cases. Since the candidates of frequent induced subgraphs are normal forms, the normalization must be applied to X_k of each transaction G to check if the candidates are contained in G. As previously described, the procedure of the normalization of X_k can derive the normal form of every induced subgraph of G in the intermediate levels. Thus, the frequency of each candidate is counted based on all normal forms of the induced subgraphs of G. When the value of the count exceeds the threshold *minsup*, the subgraph is a frequent induced subgraph. Once all frequent induced subgraphs are found, the association rules among them whose confidence values are more than a given confidence threshold are enumerated by using the algorithm similar to the standard basket analysis.

3 Performance Evaluation

The performance of the proposed AGM was examined using an artificial graph transaction data. The machine used is a PC with 400MHz CPU and 128MB main memory. The size of each transaction is determined by the gaussian distribution with the average $|T|$ and the standard deviation 1. The vertex labels are randomly determined with equal probability. The edges are attached randomly with the probability of p. L basic patterns of the average size $|I|$ are generated, and one of them is randomly overlaid on each transaction. The two groups of the test data, one for the directed graph and the other for the undirected graph, are prepared. The direction of the edges are given randomly in the former group.

Figures 1, 2, 3 and 4 show the results of computation time for different number of transactions, number of vertex labels, minimum support threshold and average transaction size for both directed and undirected graphs, respectively. In every parameter setting, the required computational time and the number of the discovered frequent induced subgraphs are less in the case of directed graph. Because the number of possible subgraph patterns is larger due to existence of edge direction, the frequency of each subgraph pattern is smaller. This also reduces the required computation for search. In short summary, the proposed algorithm does not show intractable computational complexity except the cases for graphs of large size in the database.

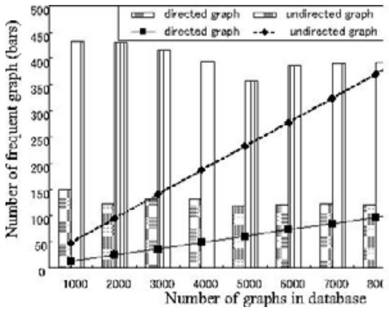

Fig. 1. Complexity v.s. number of transactions.

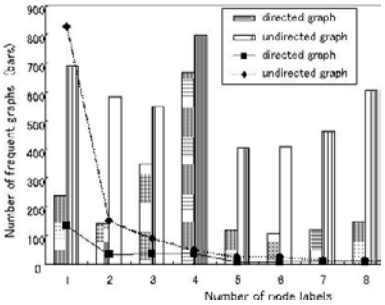

Fig. 2. Complexity v.s. number of vertex labels.

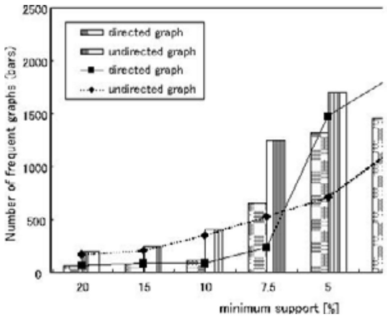

Fig. 3. Complexity v.s. minimum support.

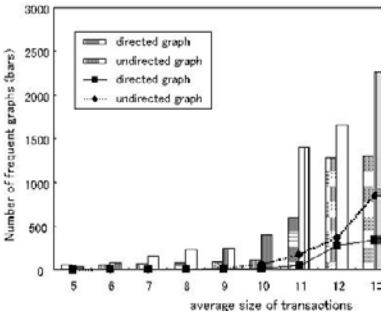

Fig. 4. Complexity v.s. transaction size.

4 Application to Chemical Analysis

AGM was applied to chemical carcinogenesis analysis which is a challenge topic proposed in IJCAI-97 by Srinisavan et al. [13]. The task is to find structures typical to carcinogen of organic chlorides. The objective data were obtained from the website of National Toxicology Program (NTP) and Oxford University. Totally, the 300 compounds were selected for the analysis, of which 185 compounds

Table 1. Results for three *minsup* values.

	minsup = 20%		*minsup* = 15%		*minsup* = 10%	
L	NOC	NOFS	NOC	NOFS	NOC	NOFS
1	24	7	24	8	24	10
2	280	62	360	67	550	108
3	2277	477	2525	640	4558	964
4	6223	2178	9709	3333	18268	5912
5	9767	4806	18740	9372	40744	19568
6	6899	4726	19813	13479	56179	37219
7	2655	2179	11989	9499	52082	41639
8	668	655	4347	4019	33208	29817
9	118	118	1212	1199	15618	15242
10	7	7	220	220	5739	5725
11	-	-	21	21	1455	1455
12	-	-	1	1	23	23
13	-	-	-	-	15	15
Total	28918	15215	68961	41858	228663	157897

L:level(number of vertices included in frequent subgraph)
NOC:number of candidates, NOFS:number of frequent graphs

have positive carcinogenesis and the rests are negative. Thus, the fraction of the carcinogenic compounds is 61.7%. The types of atoms involved in the compounds are C, H, O, Cl, F, S and some cations, and the types of bonds are single, double, aromatic and cation bonds. Each transaction data were preprocessed to add artificial edges from each vertex to every other vertex that is within the distance of 6 edges. Each added edge has a label to indicate the distance between the two vertices that are connected by the edge. This enables us to mine the frequent cooccurrence of some specific structures at a specific distance within 6. The distance limit of 6 was determined based on the chemical insights that the influence of an atom does not usually propagate along the path more than 6 bonds in molecules of moderate sizes. Furthermore, an isolated vertex labeled by the carcinogenesis class of the compound, *i.e.*, "class vertex", is added to each chemical structure graph.

The analysis was made on the same PC described in the previous section. Table 1 shows the number of the candidate induced subgraphs (NOC) and that of the discovered frequent induced subgraphs (NOF) for each level of the search, *i.e.*, the size of the induced subgraphs. In each *minsup* case, all frequent induced subgraphs were exhaustively discovered. The computation time required to complete the search was far longer for the *minsup* of smaller value, and was almost 8 days for 10%, while it was only about 40 minutes for 20%. The size of the largest frequent induced subgraph discovered in the case of 10% was 13.

In Figure 5, the confidence deviation Δ of an association rule $G_b \Rightarrow G_h$ is given as follows.

$$\Delta = \begin{cases} conf(G_b \Rightarrow G_h) - fr_p & \text{if } G_h \text{ contains a positive class vertex.} \\ conf(G_b \Rightarrow G_h) - fr_n & \text{if } G_h \text{ contains a negative class vertex.} \end{cases}$$

Here, fr_p is the fraction of positive compounds in the data, *i.e.*, 61.7% in this case, and fr_n is that of negative compounds, i.e., 38.3% (=100%-61.7%). The cover rate CR of a set of association rules is the fraction of chemical compounds whose classes are derived by applying the rule set to the data. Given a value of Δ_{th}, a set of association rules each having Δ more than the Δ_{th} is defined, and CR of the rule set is calculated. As shown in Figure 5, the rule set derived for the 10% threshold contains some rules having significant confidence. Accordingly, the exhaustive search for low support threshold is considered to be very effective to mine valuable rules.

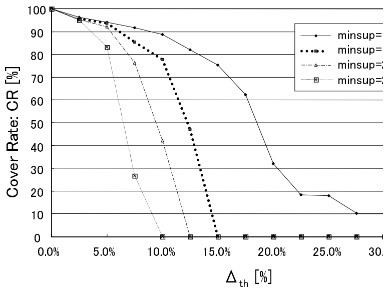

Fig. 5. Relation of Δ_{th} and CR.

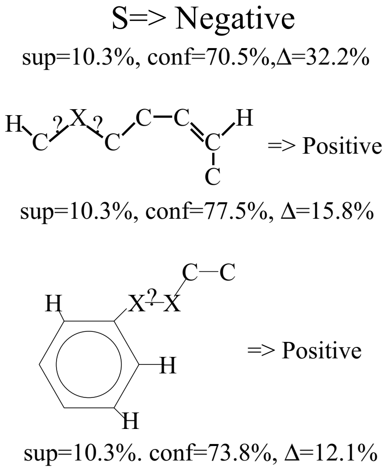
S=> Negative
sup=10.3%, conf=70.5%,Δ=32.2%
H
C
?
X
?
C
C
C
C
H
C
=> Positive
sup=10.3%, conf=77.5%, Δ=15.8%
C
C
H
X
?
X
H
H
=> Positive
sup=10.3%. conf=73.8%, Δ=12.1%

Fig. 6. Examples of discovered rules.

Figure 6 shows some association rules obtained for the carcinogenesis class under the support threshold 10%. The first rule is very simple, but indicates that a sulfur atom plays an important role to suppress the carcinogenesis. In the second rule, the symbol X of a vertex and ? of an edge indicate that their labels are arbitrary. The third is an example of a less significant but more complex substructure involving a benzene ring. This is consistent with the chemical knowledge that benzene rings frequently have the positive carcinogenesis.

5 Discussion and Conclusion

The largest graphs of the chemical compound discovered by AGM have the size of 13 atoms. In contrast, the approach of ILP in conjunction with a levelwise search proposed by Dehaspe et al. could mine the substructure consisting of 6 predicates at maximum equivalent to the size of a molecule consisting of only 3 atoms or so [3]. This fact shows the practical efficiency of AGM for real world problems. Further investigation on the computational efficiency of AGM in terms of the theoretical aspect remains for the future study.

In conclusion, a novel approach has been developed that can efficiently mine frequently appearing induced subgraphs in a given graph data set and the association rules among the frequent induced subgraphs. Its performance has been evaluated for both the artificial simulation data and the real world chemical carcinogenesis data. The powerful performance of this approach under some practical conditions has been confirmed through these evaluations.

Acknowledgement. The authors wish to thank Prof. Takashi Okada in Center for Information & Media Studies, Kwansei Gakuin University for providing us with the chemical compound data and his expertise in the chemistry domain.

References

1. Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In Proc. of the 20th VLDB Conference, pp.487–499.
2. Cook, D.J. and Holder, L.B. 1994. Substructure Discovery Using Minimum Description Length and Background Knowledge, Journal of Artificial Intelligence Research, Vol.1, pp.231-255.
3. Dehaspe, L., Toivonen, H. and King, R.D. 1998. Finding frequent substructures in chemical compounds. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), pp.30–36.
4. Fortin, S. 1996. The graph isomorphism problem. Technical Report 96-20, University of Alberta, Edomonton, Alberta, Canada.
5. Inokuchi, A., Washio, T. and Motoda, H. 1999. Derivation of the topology structure from massive graph data. Discovery Science: Proceedings of the Second International Conference, DS'99, pp.330–332.
6. Inokuchi, A. 2000. The study on a fast mining method from massive graph structure data. Master thesis (in Japanese), I.S.I.R., Osaka Univ.
7. King, R., Muggleton, S., Srinivasan, A. and Sternberg, M. 1996. Structure-activity relationships derived by machine learning; The use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. In Proceedings of the National Academy of Sciences, Vol.93, pp.438-442.
8. Klopman, G. 1984. Artificial intelligence approach to structure activity studies. J. Amer. Chem. Soc., Vol.106, pp.7315-7321.
9. Klopman, G. 1992. MultiCASE 1. A hierarchical computer automated structure evaluation program, QSAR, Vol.11, pp.176–184.
10. Kramer, S., Pfahringer, B. and Helma, C. 1997. Mining for causes of cancer: Machine learning experiments at various levels of detail. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), pp.223–226.
11. Mannila, H. and Toivonen, H. 1997. Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery, Vol.1, No.3, pp.241-258.
12. Matsuda, T., Horiuchi, T., Motoda, H. and Washio, T. 2000. Extension of Graph-Based Induction for General Graph Structured Data. In Proceedings of the Fourth Pacific-Asia Conference of Knowledge Discovery and Data Mining (PAKDD2000), pp.420–431.
13. Srinisavan, A., King, R.D., Muggleton, S.H. and Sternberg, M.J.E. 1997. The predictive toxicology evaluation challenge. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), pp.4–9.
14. Wang, K. and Liu, H. 1997. Schema discovery for semistructured data. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), pp.271–274.