Applying Algebraic Mining Method of Graph Substructures to Mutageniesis Data Analysis

Akihiro Inokuchi[1], Takashi Washio[1], Takashi Okada[2] and Hiroshi Motoda[1]

[1] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
{inokuchi,washio,motoda}@ar.sanken.osaka-u.ac.jp
[2] Center for Information & Media Studies, Kwansei Gakuin University,
1-155, Uegahara, Nishinomiya, Hyogo 662-8501, Japan
okada@kwansei.ac.jp

Working notes of Int. Workshop: KDD Challenge 2000, PAKDD2000, pp.41-46, 2000

1 Introduction

Data having graph structure are abound in many practical fields such as molecular structures of chemical compounds and information flow patterns in the internet. We have been investigating the algorithm to mine frequently occurring subgraph patterns from graph structured data. Our recent study introduced the algebraic graph theory to the framework of Basket Analysis, and achieved to extend the conventional approach to the algorithm which can efficiently mine a complete set of all frequent subgraphs from the general class of the graph structures. The graph can be either directed or undirected. It can have loops including self-loops, and the nodes and links can have labels, e.g., C (carbon) and N (nitrogen) or single bond and aromatic bond in chemical compounds. Furthermore, it can mine subgraph patterns partitioned into multiple parts. The proposed approach finds association rules in form of $G_a \Rightarrow G_b$ which represents that the occurrence of the union of the subgraphs G_a and G_b is more than a threshold support level, and the occurrence of the subgraph G_a indicates the co-occurrence of G_b with more than a threshold confidence level. Our algorithm has been applied to obtain the rules to predict the mutagenesis activity of 230 aromatic and heteroaromatic nitro compounds. Many association rules having meaningful confidence were discovered which presents the characteristic and complex substructures having either high, medium, low and inactive mutagenesis activities.

2 Algorithm

2.1 Representation of Graph Structured Data

In the framework of this paper, one graph constitutes one transaction. The graph structured data can be transformed without much computational effort into an adjacency matrix which is a very well known representation of a graph in mathematical graph theory[2]. A node which corresponds to the i-th row (the i-th column) is called the i-th node v_i and the number of node contained in a graph its "*size*". Let an adjacency matrix of a graph whose size is k be X_k, the ij-element of X_k, x_{ij} and its graph, $G(X_k)$. The node labels are defined as N_p $(p = 1, \cdots, \alpha)$ and the link labels, L_q $(q = 1, \cdots, \beta)$. Labels of nodes and links are indexed by natural numbers for the computational efficiency.

Let the set of nodes of G be $V(G)$ and the set of links of G $E(G)$. An induced subgraph G' of G is defined as follows.

$$V(G') \subset V(G),\ \ E(G') \subset E(G),\ \ \forall u, v \in V(G')\ \ \{u, v\} \in E\{G\} \Rightarrow \{u, v\} \in E\{G'\},$$

where $\{u, v\}$ represents a link to connect the nodes u and v. Based on this definition, the "*support*" of an induced subgraph $G_a \cup G_b$ of the given transactions and the "*confidence*" of an association rule $G_a \Rightarrow G_b$ are defined as follows where G_a is also an induced subgraph of $G_a \cup G_b$.

$$sup(G_a \cup G_b) = \frac{the\ \ number\ \ of\ \ transactions\ \ which\ \ include\ \ G_a \cup G_b\ \ as\ \ induced\ \ subgraph}{the\ \ number\ \ of\ \ transactions},$$

$$conf(G_a \Rightarrow G_b) = \frac{sup(G_a \cup G_b)}{sup(G_a)}.$$

Our algorithm generates association rules having support and confidence greater than user specified thresholds "*minimum support*" and "*minimum confidence*". The graph whose frequency exceeds the minimum support is called "*frequent graph*".

The code of adjacency matrices is defined as follows. In case of an undirected graph, the code $code(X_k)$ of an adjacency matrix X_k (Eq.(1)) is represented by Eq.(2) scanning the elements in the upper triangular part of X_k except its diagonal elements.

$$X_k = \begin{pmatrix} 0 & x_{1,2} & x_{1,3} & \cdots & x_{1,k} \\ x_{2,1} & 0 & x_{2,3} & \cdots & x_{2,k} \\ x_{3,1} & x_{3,2} & 0 & \cdots & x_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{k,1} & x_{k,2} & \cdots & \cdots & 0 \end{pmatrix}, \tag{1}$$

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

The code $code(X_k)$ in the case of directed graph is defined as follows.

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

2.2 Candidate Generation of Frequent Graph

Two frequent graphs are joined only when the following three constraints are satisfied to generate a candidate of frequent graph of size k+1. Let X_k and Y_k be adjacency matrices of two frequent 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}.

Constraint1

$$X_k = \begin{pmatrix} X_{k-1} & \boldsymbol{x}_1 \\ \boldsymbol{x}_2^T & 0 \end{pmatrix}, \quad Y_k = \begin{pmatrix} X_{k-1} & \boldsymbol{y}_1 \\ \boldsymbol{y}_2^T & 0 \end{pmatrix}, \quad Z_{k+1} = \begin{pmatrix} X_{k-1} & \boldsymbol{x}_1 & \boldsymbol{y}_1 \\ \boldsymbol{x}_2^T & 0 & z_{k,k+1} \\ \boldsymbol{y}_2^T & z_{k+1,k} & 0 \end{pmatrix},$$

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. Let the label of the i-th node of the adjacency matrix X_k be $N(X_k, i)$, then the following relations hold among the adjacency matrices X_k, Y_k and Z_{k+1}.

Constraint2

$$N(X_k, i) = N(Y_k, i) = N(Z_{k+1}, i), \quad N(X_k, i) \leq N(X_k, i+1), \quad i = 1, \cdots, k-1$$

$$N(X_k, k) = N(Z_{k+1}, k), \quad N(Y_k, k) = N(Z_{k+1}, k+1), \quad N(X_k, k) \leq N(Y_k, k)$$

Here, the $(k, k+1)$ and the $(k+1, k)$ elements of the adjacency matrix Z_{k+1} are not determined by X_k and Y_k. In case of an undirected graph, two possible cases are considered in which 1) there is a link labeled L_q between the k-th node and the k+1-th node of $G(Z_{k+1})$ and 2) there is no link between them. Accordingly $\beta + 1$ adjacency matrices whose $(k, k+1)$-element and $(k+1, k)$-element are "0" and "L_q" are generated. In case of a directed graph $(\beta + 1)^2$ different adjacency matrices are generated. We call X_k and Y_k the first matrix and the second matrix to generate Z_{k+1} respectively. Note that when the labels of the k-th nodes of X_k and Y_k are the same, switching X_k and Y_k, *i.e.*, taking Y_k as the first matrix and X_k as the second matrix, produces redundant adjacency matrices. In order to avoid this redundant, generation the two adjacency matrices are joined only when constraint 3 is satisfied.

Constraint3

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

We call the adjacency matrix generated under the three constraints a "*normal form*". The graph G of size k+1 is a candidate of frequent graphs only when adjacency matrices of the all induced subgraphs whose size are k are confirmed to be frequent graphs. If any of the induced subgraphs of $G(Z_{k+1})$ are not frequent graphs, Z_{k+1} is not a candidate frequent graph, because any induced subgraph of a frequent graph must be a frequent graph due to the monotonicity of the support value.

As this algorithm generates only adjacency matrices of the normal form in the earlier k-stages, if the adjacency matrix of the induced subgraph generated by removing the i-th node $(1 \leq i \leq k+1)$ is non-normal form, the transform of the matrix of normal form to that of a normal form is needed. Figure 1 shows an example of the transformation of an adjacency matrix X_4 which is non-normal form, where all the nodes and the links have a unique label. The number below each adjacency matrix in this figure shows each corresponding code respectively, and v_i denotes the i-th node of the adjacency matrix X_4 to be normalized. It starts with the adjacency matrices representing the induced subgraphs of X_4 consisting of one node (See Fig.1 A). As it is necessary to find a normal form of $G(X_4)$ among many normal forms, the combination to join should be restricted. In this case, we choose a matrix consisting of v_1 as the first matrix and the others the second matrices (See Fig.1 B). The values which can not be determined in the joining procedure are taken from the original matrix, for example x_{12} and x_{21} of X_4 are substituted to (1,2)-element and (2,1)-element of a matrix consisting of v_1 and v_2 (See Fig.1 C). Next, the matrices whose graphs have two nodes are joined (See Fig.1 D). Here, the matrix whose code is the least becomes the first matrix, and the others become the second matrices (See Fig.1 E). If there are the adjacency matrices whose codes are for tie, we simply choose one randomly. These processes continue until a normal form having the same size with the original X_4 is found. By applying the transformation to each induced subgraph obtained by removing the i-th node $(1 \leq i \leq k+1)$, the check if they are frequent graphs is easily performed.

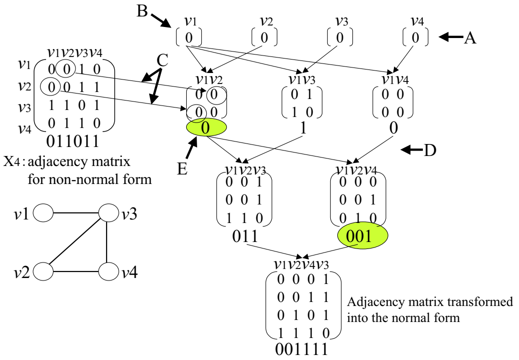

A
B
C
D
E
v1
v2
v3
v4
v1v2
v1v3
v1v4
0
1
0
v1v2v3
v1v2v4
011
001
v1v2v4v3
001111
v1v2v3v4
011011
X4: adjacency matrix for non-normal form
Adjacency matrix transformed into the normal form

Fig. 1. Transformation into the normal form

2.3 Canonical Form

After all candidate graphs are derived, it is necessary to count the support by scanning the database. However, a unique subgraph is represented by multiple adjacency matrices of normal form. Therefore, the adjacency matrices should be grouped to the corresponding unique subgraph, and if more than one occurrence of the same graph is observed in a transaction, it is counted as one in the calculation of the support value. For this grouping, the adjacency matrices representing the unique graph are indexed to a unique matrix whose code is the least among the matrices. This unique matrix is called "*canonical form*". Figure 2 shows the algorithm to obtain the canonical form of any normal form and the corresponding transformation matrix.

The matrix X^m_{k-1} made by removing the m-th node $(1 \leq m \leq k)$ from $G(X_k)$ is first transformed into the normal form by the procedure explained in Fig.1. Because this transformation is a sort procedure of the rows and columns of X^m_{k-1}, it is expressed with a transformation matrix T^m_{k-1}, and the transformed matrix of normal form is given by $(T^m_{k-1})^T X^m_{k-1} T^m_{k-1}$. We assume that a transformation matrix S^m_{k-1} from a normal form to the canonical form of each frequent graph of size k-1 is known. The matrix of canonical form of

```
forall $X_k$ in a set of candidate frequent graphs
  $X'_k = X_k$
for($m = 1; m \le k; m++$) do begin
  if($N(X_k, k) = N(X_k, m)$) then do begin
    if($code(X'_k) > code((T^m_k S^m_k)^T X_k (T^m_k S^m_k))$) then do begin
      $X'_k = (T^m_k S^m_k)^T X_k (T^m_k S^m_k)$;
      if(the canonical form of $X'_k$ is known) then do begin
        $X'_k = S'^T_k X'_k S'_k$; //where $S'_k$ is the matrix to transform $X'_k$ in r.h.s. to its canonical form
        break;
      end
    end
  end
end
if($X_k = X'_k$)
  $X'_k$=permutation($X_k$);
  if($X_k \neq X'_k$)
    If there are matrices whose canonical form is $X_k$,
    then the canonical form of these matrices is renewed to $X'_k$.;
  end
end
Canonical form of $X_k$ is $X'_k$;
end
```

Fig. 2. An algorithm to transform the canonical form

X^m_{k-1} is given by $(T^m_{k-1} S^m_{k-1})^T X^m_{k-1} T^m_{k-1} S^m_{k-1}$. The matrices S^m_k and T^m_k to transform X_k are generated from S^m_{k-1} and T^m_{k-1} by Eqs.(3) and (4).

$$s_{ij} = \begin{cases} s^m_{ij} & 0 \le i \le k-1 \text{ and } 0 \le j \le k-1, \\ 1 & i = k \text{ and } j = k, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

$$t_{ij} = \begin{cases} t^m_{ij} & i < m \text{ and } j \neq k, \\ t^m_{i-1,j} & i > m \text{ and } j \neq k, \\ 1 & i = m \text{ and } j = k, \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

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. The code of the canonical form for X_k is given by

$$\min_{m=1,\cdots,k} code((T^m_k S^m_k)^T X_k (T^m_k S^m_k)) \tag{5}$$

The matrix to transform X_k into the canonical form is $T^m_k S^m_k$ that minimizes Eq.(5).

3 Application

The algorithm explained in the aforementioned section is implemented, and the association rules have been derived from a set of mutagenesis data given in the web page of PAKDD Workshop. Mutagenesis activity is discretized into four categories according to the information described in the web page. The percentages of the transaction having the classes of high, medium, low and inactive are 15.2%, 45.7%, 29.5% and 9.6% respectively. As our algorithm can not deal with numeric features, LogP and LUMO must be discretized and labeled by some symbols. They are discretized by the method we proposed in the past [3]. It is based on AIC(Akaike Information Criterion) technique. The method discretizes the numeric features to minimize the following equation in greedy manner.

$$AIC = 2 \sum_r n(r) Ent(r) + 2m,$$

where $n(r)$ is the number of transactions located in a discretized region r of the feature space, $Ent(r)$ is the information entropy of the data in r, and m is the total number of cut points. We selected two threshold values from the cut points which are specified by this method. They are LogP=3.3 and LUMO=-1.84. As shown in Fig.3, these features of each chemical compound are added in form of isolated nodes to the transaction. Furthermore, the artificial links are added to connect each node to the other nodes where the number of links between the former node and the latter is 2 to 6.

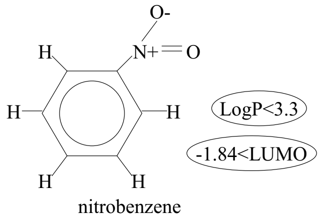
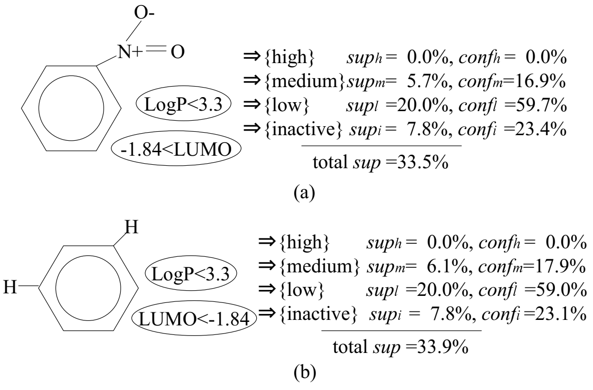


Fig. 3. Transaction

The number of frequent graphs derived for the minimum support of 20% was 64,973 in this application. Two examples among them are depicted in Fig.4. The association rules for the frequent graphs are obtained for each class of high, medium, low and inactive. The figure shows the rules where the rules for the four classes having an identical body are combined. The percentage of the transactions including this frequent graph and having each class is also indicated as sup_h, sup_m, sup_l and sup_i respectively. sup is the summation of these partial support values. The confidence value of the rule for each class is also shown respectively. The sup of the frequent graph of Fig.4(a) is 33.5%. The confidence values for the classes of low and inactive are higher than their percentages in the original distribution, i.e., 29.5% and 9.6%, while the confidence values for high and medium are lower than those percentages, i.e., 15.2% and 45.7%. This fact indicates that the substructure and the features of the chemical compound have low or negligible mutagenesis. The substructure and the features shown in Fig.4(b) indicate the similar tendency.



O-
N+
O
LogP<3.3
-1.84<LUMO
⇒{high} suph= 0.0%, confh= 0.0%
⇒{medium}supm= 5.7%, confm=16.9%
⇒{low} supl =20.0%, confl =59.7%
⇒{inactive} supi= 7.8%, confi =23.4%
total sup =33.5%
(a)
H
H
LogP<3.3
LUMO<-1.84
⇒{high} suph= 0.0%, confh= 0.0%
⇒{medium} supm= 6.1%, confm=17.9%
⇒{low} supl =20.0%, confl =59.0%
⇒{inactive} supi = 7.8%, confi=23.1%
total sup =33.9%
(b)

Fig. 4. Extracted association rules(1)

On the other hand, the substructures and the features shown in Fig.5 have higher confidence values for high and medium classes while lower values for low and inactive classes. Therefore, the compounds having these substructures and features show medium or high mutagenesis. The symbols of X and ? represent that an arbitrary atom and bond must exist at the locations respectively.

4 Related Work

The propositional classification techniques, e.g., C4.5, and the inductive logic programming (ILP) techniques have been applied to the carcinogenesis predictions of chemical compounds [4], [5]. However, these approaches can discover only limited types of characteristic substructures, because the graph structures

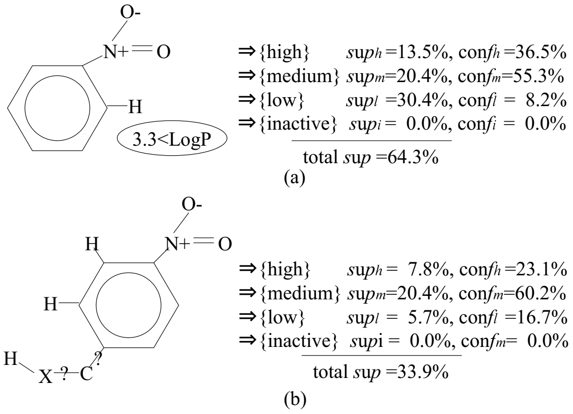

O-
N+=O
H
3.3<LogP
⇒{high} suph=13.5%, confh=36.5%
⇒{medium} supm=20.4%, confm=55.3%
⇒{low} supl =30.4%, confl = 8.2%
⇒{inactive} supi= 0.0%, confi = 0.0%
total sup =64.3%
(a)
O-
H
N+=O
H
H
X ? C
?
⇒{high} suph= 7.8%, confh=23.1%
⇒{medium} supm=20.4%, confm=60.2%
⇒{low} supl = 5.7%, confl =16.7%
⇒{inactive} supi = 0.0%, confm= 0.0%
total sup =33.9%
(b)

Fig. 5. Extracted association rules(2)

must be predefined by some specific features and/or ground instances of predicates such that a benzene ring is involved in the compound. This data preprocessing is inevitably needed for the propositional classification techniques, since they can handle only feature tables. This preprocessing is also necessary for ILP techniques to reduce the computation time in the mining process. However, our algorithm can directly handle the graph structure of general class.

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. [6]. They used the framework of the ILP combining levelwise search to minimize the access frequency to the database. Since the efficiency achieved by this approach is better to the former ILP approaches, some discovery of substructures characterizing carcinogenesis was expected. However, the full search space was still so large that the search had to be limited to within the 6th level where the substructures consist of a few atoms at maximum, and they reported that significant substructures have not been obtained within the search level.

5 Conclusion

By applying the mining method of substructures based on the algebraic graph theory, many association rules having meaningful confidence were discovered in the analysis of mutagenesis data. The rule represents the characteristic and complex substructures having either high, medium, low and inactive mutagenesis activities.

References

1. R. Agrawal and R. Srikant.: Fast algorithms for mining association rules. *Proc. of the 20th VLDB Conference*, pp. 487-499, 1994.
2. N. Biggs.: Algebraic Graph Theory. Cambridge Univ. Press. (1973)
3. A. Inokuchi, T. Washio and H. Motoda.: Basket Analysis for Numerical Attribute Data(in Japanese). *Proc. of the 12th Annual Coference of JSAI*, pp. 74-76, 1998.
4. Kramer, S., Pfahringer, B. and Helma, C.: Mining for causes of cancer: Machine learning experiments at various levels of detail. *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 223-226, 1997.
5. King, R., Muggleton, S., Srinivasan, A. and Sternberg, M.: Structure-activity relationships derived by machine learning .*Proc. of the National Academy of Sciences*, Vol.93, pp. 438-442, 1996.
6. Dehaspe, L., Toivonen, H. and King, R.D.: Finding frequent substructures in chemical compounds. *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining*, pp. 30-36, 1998.