A Study On Data Mining Algorithm Gragh-based

Tang Qiuyu
Computer and Information Engineering Dept.
Baoding Vocational and Technical College
Baoding City, China,13930828561
Tang_qy@126.com

Tang Yifan
Beijing Institute of Technology
Beijing City, China,13303121998
Tang_yifan@163.com

***Abstract*—In computer science, graph is one of the most complex data structures. But graph can be used in research and in bussiness because of its intuitive expression. It is a focus in data mining how to derive the interesting subgraph patterns from a great number of graphs. This article introduces a data mining algorithm based on graph which ueses the graph theory to improve the classic Apriori algorithm. And this algorithm can mine frequent subgraph patterns effectively.**

Keywords- data mining; algorithm; graph; Apriori

I. INTRODUCTION

Data mining is to mine some useful knowledge or information from a large number of randam data. The data can be structured or semi-structured, even can be heterogeneous. This paper proposes a algorithm used in graph-based data mining which is a kind of semi-structured one.

To compare with the general data mining algorithm, graphs can express richer semantic and can be more widely used in research and business.

II. THE PRINCIPLE OF GRAPG-BASED DATA MINING

The graph-based data mining is mainly to find frequent subgraph, whose support degree is more than the min-degree, in data base. Apriori is a classical algorithm which can mine frequent itemsets in affairs data base effectively. But because of the complex structure of graph, the original Apriori algorithm can not be used in graph-based data mining. We have to improve Apriori to adapt mining of the frequent subgraph.

We have to solve the isomorphism problem before mining frequent subgraph. In intuitive, the problem is to determine if graph G' is subgraph of graph G. The isomorphism problem of subgraph is proved to be a NP one. So we have to use additional restrictions to reduce the searching space and time complexity of algorithm.

A. Definitions

Definition 1: Subgraph isomorphism. Graph G' is isomorphic to graph G. If and only if G has a subgraph G'', G'' is isomorphic to G'.

Definition 2: Induced subgraph. V(G) is set of vertices of graph G and E(G) is set of edges of G. G' is the induced subgraph of G. If and only if $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$ and $\forall u$, $v \in V(G')$, $\{u,v\} \in E(G) \Leftrightarrow \{u,v\} \in E(G')$. In intuitive, the set of vertices of G', the induced subgraph of G, is a subset of that of G and the set of edges is determined by G. The subgraph mining that metioned in this paper is the set of induced subgraph.

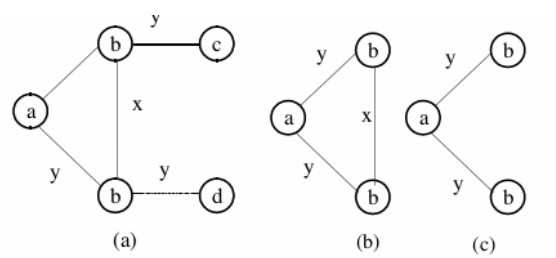

Figure 1. Subgraph isomorphism And Induced subgraph

In Fig.1 (b) is a induced subgraph of Fig.1 (a). And Fig.1 (b) is isomorphic to Fig.1 (a). Fig.1 (c) is a general subgraph of Fig.1 (a).

After the introduction of two definitions, we shall mark all the vertexes and edges of the graph to distinguish them. At the same time, we shall allow the same vertexes mark and edges mark in the same graph. After the marking of verteses and edges, we can reduce the searching space greatly at the time of subgraph matching.

The data base D={G0,G1,G2, ⋯, Gn}. The support dagree of induced subgraph can be definited as the follow.

$$\mathrm{Sup(G)} = \frac{N1}{N2}$$

N1 is the number of affairs of G including in D. N2 is the number of affairs of D. The frequent induced subgraph is the induced subgraph whose support degree is more than min-sup. Size(G) is the number of vertexes of G.

B. Analysis of Data Structure

The data structure of undirected graph can be definited as a adjacency matrix. According to symmetry, we only retain the lower triangular matrix of the adjacency matrix to improve the efficiency of algorithm. So the adjacency matrix $x_{i,j}$ of undirected graph G can be definited as the follow.

$$xi,j=\begin{cases}0, \text{if } i<j\\ \text{the mark of vertex } L(vi), \text{if } i=j\\ \text{the mark of edge } L(\{vi,vj\}), \text{if } i>j\end{cases}$$

That is to say that diagonal elements are the mark of vertex and other elements in lower triangular are the mark of edge or 0. In the same time, the mark of vertex is ordered by dictionary. The adjacency matrix of Fig.1 (a) can be represented as Fig.2. Even if the mark of vertex is ordered by dictionary, we can not ensure the uniqueness of adjacency matrix because the appearance of same marks. So we lead normal matrix into this paper to ensure the correspondence between matrix and graph. Firstly, we can encode the adjacency matrix X which meets the above definition.

$$Code(X)=x_{11}x_{21}x_{22}\cdots x_{n1}x_{n2}\cdots x_{n,n-1}x_{nn}$$

The normal matrix can be definited as the follow. When graph G only has one adjacency matrix, the adjacency matrix is the normal matirx of G. And when G has more than one adjacency matrix, the adjacency matrix, which has a maximum of code, is the normal matrix of G. The value of code can also be ordered by dictionary (a>b>c> ··· >z>0). If the two adjacency matrixes M1 and M2 are as Fig.2, their values of code are:

Code(M1)= aybyxb0y0c00y0d

Code(M2)= aybyxb00yc0y00d

Because Code(M1)>Code(M2), M1 is the normal matrix of Fig.1 (a).

a				
y	b			
0	y	0	c	
0	0	y	0	d

a				
y	b			
y	x	b		
0	0	y	c	
0	y	0	0	d

Figure 2. Fig 1 (a) of the adjacency matrix

C. *Improved Data Mining Algorithm*

Apriori algorithm searches the frequent itemsets with the candidate sets. The candidate sets generate from bottom to top. That is to say the candidate k+1 itemset generates from frequent k itemset. It has two main steps: connection and pruning. In the steps, we can use the character of Apriori algorithm to reduce the research space. The character of Apriori algorithm is that all the nonempty subsets of frequent itemsets are frequent. If we expand the character to graph-based data base, all the nonempty induced subgraphes of frequent induced subgraphes are frequent.

Step 1. Connection. Assuming matrix X_k and Y_k are normal matrixes, which has k vertexes, corresponding to the frequent induced subgraphes.

$X_k=\begin{pmatrix} x_{k-1},0 \\ x^T, x_{kk}\end{pmatrix}$, $Y_k=\begin{pmatrix} x_{k-1},0 \\ y^T, y_{kk}\end{pmatrix}$. If we connect X_k with Y_k, we can generate a matrix Z_{k+1}, which has k+1 vertexes, corresponding to candidate induced subgraph.

$$Z_{k+1}=\begin{pmatrix} x_{k-1},0 \\ X^T, X_{kk} \\ y^T Z_{k+1k} y_{kk}\end{pmatrix}$$

In the above formula, x^T and y^T are n-dimensional vectors. The value of $Z_{k+1,k}$ is determined by whether there is an edge between x_{kk} and y_{kk}. And it has two possibilities. Commonly, the two are used dircectly. But we improve this method in this paper. We determine the value of $Z_{k+1,k}$ in the following way. If there is no edges between x_{kk} and y_{kk}, $Z_{k+1,k}$=0 and there is only one result of connection. If there are edges, $Z_{k+1,k}$=mark of this edge and there is two results of connection. If the subgraph corresponding to Z_{k+1} is frequent, all its induced subgraph are frequent because of the character of Apriori. So if there are edges between x_{kk} and y_{kk}, the edges are frequent. We can get the value of mark of edges by scaning the induced frequent subgraphes of 2-itemsets. Fig. 3 is a connecting example. In Fig.3 whether there are edges between b and c is determined by frequent 2-itemsets.

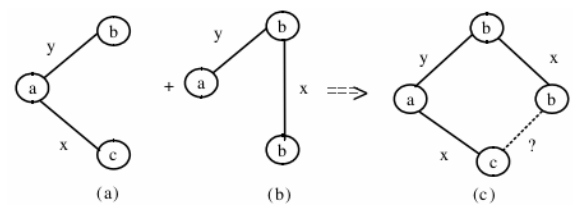

Figure 3. A connection instance

We can get a matrix after connection. But this matrix might not a normal one. So we have to adjust it into normal one.

Step 2. Pruning. The goal of this step is to reduce the candidate space. That is to inspect wheathea Z_{k+1} is the candidate of the frequent induced subgraphes. Because of the character of Apriori, we can inspect all the k-item subsets Z_k of Z_{k+1}. If and only if all the Z_k belong to the candidates of k-item frequent induced subgraphes, we can add Z_{k+1} into k+1-item candidate of frequent induced subgraphes.

Step 3. Calculating the support degree of Z_{k+1}. After the generation of k+1-item candidate of frequent induced subgraphes, we can calculate the support degree of every candidate Z_{k+1} with scaning data base. This step is to determine whether it is the frequent induced subgraph. When Z_{k+1} corresponds to the mark of vertexes and edges of G'. G' is the subgraph of some affair G, we can add 1 to the value of this candidate. After scan, we can get the support degree with the value of candidate dividing by the number of affairs in data base. If the support degree is more than the min-support degree, we can add Z_{k+1} into k+1-item frequent induced subgraph.

III. DESCRIPTION OF ALGORITHM

Input: The data base which is described with normal matrix D={M1,M2, ⋯, Mn} and min-sup.

Output: The frequent induced subgraph item L.

The main step is the same as Apriori. But we change one step in this paper. When we scan the data base, we can not only get L1, but also L2. In this way, the circle begins with searching the frequent 3-itemset.

The following is a description of the key subroutine.

A. *The Subroutine Generating C_k*

1) for each $l_1 \in L_{k-1}$

2) for each $l_2 \in L_{k-1}$//the different two in frequent k-1-itemset L_{k-1} //connect only once to avoid repeat

3) {

4) if(l1[1][1]=l2[1][1]) ∧ (l1[2][1]=l2[2][1]) ∧ (l1[2][2]=l2[2][2]) ∧ ⋯ ∧ (l1[k-2][1]=l2[k-2][1]) ∧ l1[k-2][2]=l2[k-2][2])∧⋯∧(l1[k-2][k-2]=l2[k-2][k-2])

5) {

6) Scan the frequent 2-itemset L_2. Determine wether there are edges mark between vertex of l_1[k-1][k-1] and that of l_2[k-1][k-1]. If there is not the mark, connect l_1 and l_2. Then get candidate c_1. In c_1, c1[k][k-1]=0. The other elements determine with l_1 and l_2. If there is the mark p， connect l_1 and l_2. Then get candidate c1 and c2. In it, c_1[k][k-1]=0, c_2[k][k-1]=p. The other elements of c_1 and c_2 determine with l_1 and l_2;

7) Adjust the normal matrix generating in step 6). Or c_1 and c_2 are normal matrix ;

8) For each k-1-item subsets of c_1 or c_2(when c_2 exists) //pruning

9) {

10) Adjust s to normal matrix;

11) if (s does not belong to L_{k-1})

12) Delete c_1 or c_2, go to 2;

13) }

14) Add c_1 or c_2 to C_k;

15) }//end if

16) }

B. *The Subroutine Determining Wether $c \subset t$*

1) x=c[1][1], n1=size(c), n2=size(t);// n1 and n2 are the number of vertexes

2) Traverse the diagonal elements of t

3) if (exist t[k][k]=x) goto 5);

4) return FALSE;

5) if (n2<n1+k－1) return FALSE;

6) else if (c[1][1]=t[k][k]) ∧ (c[2][1]=t[k+1][k]) ∧ (c[2][2]=t[k+1][k+1]) ∧ ⋯ ∧ (c[n1][1]=t[k+n1-1][k]) ∧ (c[n1][2]=t[k+n1-1][k+1]) ∧ ⋯ ∧ (c[n1][n1]=t[k+n1-1][k+n1-1]) // Determine wether the mark of vertexes is the same as that of edges

7) return TRUE;

8) else return FALSE;

IV. SIMULATION AND ANALYSE

In this paper, we test the performance of algorithm with a simulating experiment to a data base. Each graph elements stores in normal matrix in this data base. It includes 300 randam data. For each data, the max number of vertexes is 10 and that of edges is 20. The total number of mark of vertexes and edges are both 100.

Enviroment of experiment. Pentium Ⅲ 1GHz Cpu. 512M memory. WindowXP OS. The result of experiment is as the following Table 1.

TABLE I. THE EXPERIMENTAL RESULTS

Support parameters（%）	Mining the number of sub-graph	Running time（S）	Memory consumption（MB）
25	0	0.8	10.4
20	6	1.5	11.7
15	11	1.6	14.5
10	20	2.8	16.0
5	45	11.5	28.6
2.5	89	18.1	50.2

We can conclude from Table 1 that the running time and memory consumption increase sharply with the decrease of support degree when the support degree is small.

V. CONCLUSION

This paper is based on classical Apriori algorithm and uses adjacency matrix to represent graph. We improve some key steps: connection and pruning. In the same time, we propose a algorithm by which we can solve the problem of Isomorphism of subgraphes. Thereby, we get a efficient graph-based data mining algorithm.

REFERENCES

[1] WANG Ying-long1,YANG Jun2,ZHOU Fa-guo1,TANG Jian-jun21.School of Software,Jiangxi Agriculture University,Nanchang 330045,China 2.College of Computer and Information Engineering,Jiangxi Agriculture University,Nanchang 330045,China;Research on algorithm for mining weighted maximal frequent subgraphs[J];Computer Engineering and Applications;2009-20

[2] HUANG Jian-ming,ZHAO Wen-jing,WANG Xing-xing(School of Information and Control Engineering,Xi'an University of Architecture and Technology,Xi'an 710055);Improved Apriori Algorithm Based on Across Linker[J];Computer Engineering;2009-02

[3] CHEN Li-ning,LUO Ke(College of Computer and Communication Engineering,Changsha University of Science and Technology,Changsha Hunan 410076,China);Fast AGM algorithm and application to three-

dimensional structure analysis[J];Journal of Computer Applications;2010-12

[4] XIE Di,SHANG Xue-qun,WANG Miao,ZHANG Yan-yuanSchool of Computer,Northwestern Polytechnical University,Xi'an 710072,China;Algorithm considering imbalance across datasets for mining frequent subgraphs[J];Computer Engineering and Applications;2008-36

[5] LI Yu-hua,LUO Han-guo,SUN Xiao-lin (School of Computer Science and Technology,Huazhong University of Science and Technology,Wuhan 430074,China);A Frequent Subgraph Discovery Algorithm Based on Apriori's Idea[J];Computer Engineering & Science;2007-04

[6] CHEN An-Long1+, TANG Chang-Jie1, TAO Hong-Cai2, YUAN Chang-An1,3, XIE Fang-Jun1 1(College of Computer Science and Engineering, Sichuan University, Chengdu 610064, China) 2(College of Computer and Communication Engineering, Southwest Jiaotong University, Chengdu 610031, China) 3(Department of Information Technology, Guangxi Teachers?College, Nanning 530001, China);An Improved Algorithm Based on Maximum Clique and FP-Tree for Mining Association Rules[J];Journal of Software;2004-08

[7] YUAN Wan-lian1,2,ZHENG Cheng1,ZHAI Ming-qing2 (1.School of Computer Science and Technology,Anhui University,Hefei 230039,China;2.Department of Mathematics,Chuzhou University,Chuzhou 239012,China);An Improvement on Apriori Algorithm[J];Computer Technology and Development;2008-05

[8] WU Lei,HE Jia(School of Computers,CUIT,Chengdu 610225,China);Research of AprioriHybral algorithm based on item sets matrix[J];Journal of Chengdu University of Information Technology;2009-01

[9] Yao Qingshan1,Zhang Chunxia2,3(1.Henan Engineering Institute,Zhengzhou 451191,China; 2.Communication Vocational Technical College of Henan,Zhengzhou 450005,China; 3.Wuhan University,Wuhan 430072,China);Web Usage Mining System Based on Association Rule[J];Henan Science;2008-03

[10] Zhang Meifeng 1 Zhang Jianwei 1 Zhang Xinjing 2 Lou Shuqin 31 (Department of Computer Science and Engineering,Zhengzhou Institute of Light Industry,Zhengzhou450002) 2 (Department of Applied Mathematics and Physics,Zhengzhou Institute of Light Industry,Zhengzhou450002) 3 (School of Electronics and Information Engineering,Northern Jiaotong University,Beijing100044);Research on an Algorithm for Mining of Efficient Association Rules Based on Apriori[J];Computer Engineering and Applications;2003-19