A Correspondence Between Maximal Complete Bipartite Subgraphs And Closed Patterns

Jinyan Li, Haiquan Li, Donny Soh, Limsoon Wong
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
Email: {jinyan, haiquan, studonny, limsoon}@i2r.a-star.edu.sg

Abstract

For an undirected graph G without self-loop, we prove: (i) that the number of closed patterns in the adjacency matrix of G is even; (ii) that the number of the closed patterns is precisely double the number of maximal complete bipartite subgraphs of G; (iii) that for every maximal complete bipartite subgraph, there always exists a unique and distinct pair of closed patterns that matches the two vertex sets of the subgraph. Therefore, we can efficiently enumerate all maximal complete bipartite subgraphs by using algorithms for mining closed patterns which have been extensively studied in the data mining field.

1 Introduction

Interest in graphs and their applications has grown exponentially in the past two decades (Gross & Yellen, 2004; Makino & Uno, 2004), largely due to the usefulness of graphs as models in many areas such as mathematical research, electrical engineering, computer programming, business administration, sociology, economics, marketing, biology, and networking and communications. In particular, many problems can be modelled with *maximal complete bipartite subgraphs* (see the definition below) formed by grouping two non-overlapping subsets of vertices of a certain graph that show a kind of full connectivity between them.

We consider two examples. Suppose there are p customers in a mobile communication network. Some people have a wide range of contact, while others have few. Which groups of customers (with a maximal number) have a full interaction with another group of customers, a problem similar to one (Murata, 2004) studied in web mining? This situation can be modelled by a graph where a mobile phone customer is a node and a communication is an edge. Thus, a maximal bipartite subgraph of this graph corresponds to two groups of customers between whom there exist a full communication. Our second example is about proteins' interaction in a cell. There are usually thousands of proteins in a cell that interact with one another. This situation again can be modelled by a graph, where a protein is a node and a pair of interacting proteins forms an edge. Then, listing all maximal complete bipartite subgraphs from this graph can answer questions such as which two protein groups have a full interaction, which is a problem studied in biology (Reiss & Schwikowski, 2004; Tong et al., 2002).

Listing all maximal complete bipartite subgraphs is studied theoretically in (Eppstein, 1994). The result is that all maximal complete bipartite subgraphs of a graph can be enumerated in time $O(a^3 2^{2a} n)$, where a is the arboricity of the graph and n is the number of vertices of the graph. Even though the algorithm has a linear complexity, it is not practical for large graphs due to the large constant overhead (a can easily be around 10-20 in practice) (Zaki & Ogihara, 1998). In this paper, we study this problem from data mining perspective: We use a heuristics data mining algorithm to efficiently enumerate all maximal complete bipartite subgraphs from a large graph. A main concept of the data mining algorithm is called *closed patterns*. There are many recent algorithms and implementations devoted to the mining of closed patterns from the so-called

transactional databases (Bastide et al., 2000; Goethals & Zaki, 2003; Grahne & Zhu, 2003; Uno et al., 2004; Pan et al., 2003; Pasquier et al., 1999; Pei et al., 2000; Wang et al., 2003; Zaki & Hsiao, 2002). The data structures are efficient and the mining speed is tremendously fast. Our main contribution here is the observation that the mining of closed patterns from the adjacency matrix of a graph, termed a special transactional database, is equivalent to the problem of enumerating all maximal complete bipartite subgraphs of this graph.

The rest of this short paper is organized as follows: Sections 2 and 3 provide basic definitions and propositions on graphs and closed patterns. In Section 4 we prove that there is a one-to-one correspondence between closed pattern pairs and maximal complete bipartite subgraphs for any simple graph. In Section 5, we present our experimental results on a proteins' interaction graph. Section 6 discusses some other related work and then concludes this paper.

2 Maximal Complete Bipartite Subgraphs

A **graph** $G = \langle V^G, E^G \rangle$ is comprised of a set of vertices V^G and a set of edges $E^G \subseteq V^G \times V^G$. We often omit the superscripts in V^G, E^G and other places when the context is clear. Throughout this paper, we assume G is an undirected graph without any self-loops. In other words, we assume that (i) there is no edge $(u, u) \in E^G$ and (ii) for every $(u, v) \in E^G$, (u, v) can be replaced by (v, u)—that is, (u, v) is an unordered pair.

A graph H is a **subgraph** of a graph G if $V^H \subseteq V^G$ and $E^H \subseteq E^G$. A graph G is **bipartite** if V^G can be partitioned into two non-empty and non-intersecting subsets V_1 and V_2 such that $E^G \subseteq V_1 \times V_2$.This bipartite graph G is usually denoted by $G = \langle V_1 \cup V_2, E^G \rangle$. Note that there is no edge in G that joins two vertices within V_1 or V_2. G is **complete bipartite** if $V_1 \times V_2 = E^G$.

Two vertices u, v of a graph G are said to be adjacent if $(u, v) \in E^G$—that is, there is an edge in G that connects them. The **neighborhood** $\beta^G(v)$ of a vertex v of a graph G is the set of all vertices in G that are adjacent to v—that is, $\beta^G(v) = \{u \mid (u, v) \text{ or } (v, u) \in E^G\}$. The neighborhood $\beta^G(X)$ for a non-empty subset X of vertices of a graph G is the set of common neighborhood of the vertices in X—that is, $\beta^G(X) = \cap_{x \in X} \beta^G(x)$.

Note that for any subset X of vertices of a graph G such that X and $\beta^G(X)$ are both non-empty, it is the case that $H = \langle X \cup \beta^G(X), X \times \beta^G(X) \rangle$ is a complete bipartite subgraph of G. Note also it is possible for a vertex $v \notin X$ of G to be adjacent to every vertex of $\beta^G(X)$. In this case, the subset X can be expanded by adding the vertex v, while maintaining the same neighborhood. Where to stop the expansion? We use the following definition of maximal complete bipartite subgraphs.

Definition 2.1 *A graph $H = \langle V_1 \cup V_2, E \rangle$ is a* **maximal complete bipartite subgraph** *of G if H is a complete bipartite subgraph of G such that $\beta^G(V_1) = V_2$ and $\beta^G(V_2) = V_1$.*

Not all maximal complete bipartite subgraphs are equally interesting. Recall our earlier motivating example involving the customers in a mobile communication network. We would probably not be very interested in two groups of customers between whom there exist a full communication, if the groups both comprise a single person. In contrast, we would probably be considerably more interested if one of the group is large, or both of the groups are large. Hence, we can introduce the notion of density on maximal complete bipartite subgraphs.

Definition 2.2 *A maximal complete bipartite subgraph $H = \langle V_1 \cup V_2, E \rangle$ of a graph G is said to be (m, n)-dense if $|V_1|$ or $|V_2|$ is at least m, and the other is at least n.*

A complete bipartite subgraph $H = \langle V_1 \cup V_2, E \rangle$ of G such that $\beta^G(V_1) = V_2$ and $\beta^G(V_2) = V_1$ is maximal in the sense that there is no other complete bipartite subgraph $H' = \langle V_1' \cup V_2', E' \rangle$ of G with $V_1 \subset V_1'$ and $V_2 \subset V_2'$ such that $\beta^G(V_1') = V_2'$ and $\beta^G(V_2') = V_1'$. To appreciate this notion of maximality, we prove the proposition below.

Proposition 2.3 *Let $H = \langle V_1 \cup V_2, E\rangle$ and $H' = \langle V_1' \cup V_2', E'\rangle$ be two maximal complete bipartite subgraphs of G such that $V_1 \subseteq V_1'$ and $V_2 \subseteq V_2'$. Then $H = H'$.*

Proof: *Suppose $H = \langle V_1 \cup V_2, E\rangle$ and $H' = \langle V_1' \cup V_2', E'\rangle$ are two maximal complete bipartite subgraphs of G such that $V_1 \subseteq V_1'$ and $V_2 \subseteq V_2'$. Since $V_1 \subseteq V_1'$ and $V_2 \subseteq V_2'$, we have $\beta^G(V_1') \subseteq \beta^G(V_1)$ and $\beta^G(V_2') \subseteq \beta^G(V_2)$. Using the definition of maximal complete bipartite subgraphs, we derive $V_2' = \beta^G(V_1') \subseteq \beta^G(V_1) = V_2$ and $V_1' = \beta^G(V_2') \subseteq \beta^G(V_2) = V_1$. Then $E = V_1 \times V_2 = V_1' \times V_2' = E'$. Thus $H = H'$ as desired.* □

3 Closed Patterns of an Adjacency Matrix

The adjacency matrix of a graph is important in this study. Let G be a graph with $V^G = \{v_1, v_2, \ldots, v_p\}$. The **adjacency matrix** $\mathbf{A}$ of G is the $p \times p$ matrix defined by

$$\mathbf{A}[i,j] == \begin{cases} 1 & \text{if } (v_i, v_j) \in E^G \\ 0 & \text{otherwise} \end{cases}$$

Recall that our graphs do not have self-loop and are undirected. Thus $\mathbf{A}$ is a symmetric matrix and every entry on the main diagonal is 0. Also, $\{v_j \mid \mathbf{A}[k,j] = 1, 1 \le j \le p\} = \beta^G(v_k) = \{v_j \mid \mathbf{A}[j,k] = 1, 1 \le j \le p\}$.

The adjacency matrix of a graph can be interpreted into a **transactional database** (*DB*), which is a concept used very often in the data mining community. To define a *DB*, we first define a **transaction**. Let I be a set of **items**. Then a transaction is defined as a subset of I. For example, assume I to be all items in a supermarket, a transaction by a customer is the items that the customer bought. A *DB* is a non-empty set of transactions. Each transaction T in a *DB* is assigned a unique identity $id(T)$. A **pattern** is defined as a non-empty set[1] of items of I. A pattern may be or may not be contained in a transaction. Given a *DB* and a pattern P, the number of transactions in *DB* containing P is called the **support** of P, denoted $sup^{DB}(P)$. In this paper, unless mentioned otherwise, we consider only patterns that appear in a given transactional database *DB*. In fact, for data mining, we are often interested only in patterns that appear sufficiently frequent. That is, we consider only patterns P satisfying $sup^{DB}(P) > ms$, for a threshold $ms > 0$. Unless mentioned otherwise, we set $ms = 1$ in this paper.

Let G be a graph with $V^G = \{v_1, v_2, \ldots, v_p\}$. If each vertex in V^G is defined as an item, then the neighborhood $\beta^G(v_i)$ of v_i is a transaction. Thus, $\{\beta^G(v_1), \beta^G(v_2), \ldots, \beta^G(v_p)\}$ is a *DB*. Such a special *DB* is denoted by DB_G. The identity of a transaction in DB_G is defined as the vertex itself—that is, $id(\beta^G(v_i)) = v_i$. Note that DB_G has the same number of items and transactions. Note also that $v_i \notin \beta^G(v_i)$ since we assume G to be an undirected graph without self-loop.

DB_G can be represented as a binary square matrix. This binary matrix $\mathbf{B}$ is defined by

$$\mathbf{B}[i,j] = \begin{cases} 1 & \text{if } v_j \in \beta^G(v_i) \\ 0 & \text{otherwise} \end{cases}$$

Since $v_j \in \beta^G(v_i)$ iff $(v_i, v_j) \in E^G$, it can be seen that $\mathbf{A} = \mathbf{B}$. So, "a pattern of DB_G" is equivalent to "a pattern of the adjacency matrix of G".

Closed patterns are a type of interesting patterns in a *DB*. In the last few years, the problem of efficiently mining closed patterns from a large *DB* has attracted a lot of researchers in the data mining community (Bastide et al., 2000; Goethals & Zaki, 2003; Grahne & Zhu, 2003; Uno et al., 2004; Pan et al., 2003; Pasquier et al., 1999; Pei et al., 2000; Wang et al., 2003; Zaki & Hsiao, 2002). Let I be a set of items, and D be a transactional database defined on I. For a pattern $P \subseteq I$, let $f^D(P) = \{T \in D \mid P \subseteq T\}$—that is, $f^D(P)$ are all transactions in D containing the pattern P. For a set of transactions $D' \subseteq D$, let $g(D') = \bigcap_{T \in D'} T = \bigcap D'$—that is, the set of items which are shared by all transactions in D'. Using these

[1] The $\emptyset$ is usually defined as a valid pattern in the data mining community. However, in this paper, to be consistent to the definition of $\beta^G(X)$, it is excluded.

two functions, we can define the notion of **closed patterns**. For a pattern P, $CL^D(P) = g(f^D(P))$ is called the **closure** of P. A pattern P is said to be **closed** with respect to a transactional database D iff $CL^D(P) = P$.

We define the **occurrence set** of a pattern P in DB as $occ^{DB}(P) = \{id(T) \mid T \in DB, P \subseteq T\} = \{id(T) \mid T \in f^{DB}(P)\}$. It is straightforward to see that $id(T) \in occ^{DB}(P)$ iff $T \in f^{DB}(P)$. There is a tight connection between the notions of neighbourhood in a graph G and occurrence in the corresponding transactional database DB_G.

Proposition 3.1 *Given a graph G and a (non-empty) pattern P that occurs at least ms times in DB_G. Then $occ^{DB_G}(P) = \beta^G(P)$. Note that we do not require $occ^{DB_G}(P)$ to occur at least ms times in DB_G.*

Proof: *If $v \in occ(P)$, then v is adjacent to every vertex in P. Therefore, $v \in \beta(v')$ for each $v' \in P$. That is, $v \in \bigcap_{v' \in P} \beta(v') = \beta(P)$.*

If $u \in \beta(P)$, then u is adjacent to every vertex in P. So, $\beta(u) \supseteq P$. Therefore, $\beta(u)$ is a transaction of DB_G containing P. So, $u \in occ(P)$. □

There is also a nice connection between the notions of neighborhood in a graph and that of closure of patterns in the corresponding transactional database.

Proposition 3.2 *Given a graph G and a (non-empty) pattern P that occurs at least ms times in DB_G. Then $\beta^G(\beta^G(P)) = CL^{DB_G}(P)$. Thus $\beta^G \circ \beta^G$ is a closure operation on patterns that occur at least ms times in DB_G.*

Proof: *By construction, $\beta(\beta(P)) = \beta(occ(P)) = \bigcap_{id(T) \in occ(P)} T = \bigcap_{T \in f(P)} T = g(f(P)) = CL(P)$.* □

We discuss in the next section deeper relationships between the closed patterns of DB_G and the maximal complete bipartite subgraphs of G.

4 Results

The occurrence set of a closed pattern C in DB_G plays a key role in the maximal complete bipartite subgraphs of G. We introduce below some of its key properties.

Proposition 4.1 *Let G be a graph. Let C_1 and C_2 be closed patterns that appear at least ms times in DB_G. Then $C_1 = C_2$ iff $occ^{DB_G}(C_1) = occ^{DB_G}(C_2)$.*

Proof: *The left-to-right direction is trivial. To prove the right-to-left direction, let us suppose that $occ(C_1) = occ(C_2)$. It is straightforward to see that $id(T) \in occ(P)$ iff $T \in f(P)$. Then we get $f(C_1) = f(C_2)$ from $occ(C_1) = occ(C_2)$. Since C_1 and C_2 are closed patterns of DB_G, it follows that $C_1 = g(f(C_1)) = g(f(C_2)) = C_2$, and finishes the proposition.* □

Proposition 4.2 *Let G be a graph and C a closed pattern that occurs at least ms times in DB_G. Then C and its occurrence set has empty intersection. That is, $occ^{DB_G}(C) \cap C = \{\}$.*

Proof: *Let $v \in occ(C)$. Then v is adjacent to every vertex in C. Since we assume G is a graph without self-loop, $v \notin C$. Therefore, $occ^{DB_G}(C) \cap C = \{\}$.* □

In fact this proposition holds for any pattern P, not necessarily a closed pattern C.

Lemma 4.3 *Let G be a graph. Let C be a closed pattern that occurs at least ms times in DB_G. Then $f^{DB_G}(occ^{DB_G}(C)) = \{\beta^G(c) \mid c \in C\}$.*

Proof: *As C is a closed pattern, by definition, then $\{c \mid c \in C\}$ are all and only items contained in every transaction of DB_G that contains C. This is equivalent to that $\{c \mid c \in C\}$ are all and only vertices of G that are adjacent to every vertex in $occ(C)$. This implies that $\{\beta(c) \mid c \in C\}$ are all and only transactions that contain $occ(C)$. In other words, $f(occ(C)) = \{\beta(c) \mid c \in C\}$.* □

Proposition 4.4 *Let G be a graph and C a closed pattern that occurs at least ms times in DB_G. Then $occ^{DB_G}(C)$ is a closed pattern of DB_G.*

Proof: *By Lemma 4.3, $f(occ(C)) = \{\beta(c) \mid c \in C\}$. So $CL(occ(C)) = g(f(occ(C))) = \bigcap f(occ(C)) = \bigcap_{c \in C} \beta(c) = \beta(C) = occ(C)$. Thus $occ(C)$ is a closed pattern.* □

The three propositions above give rise to a couple of interesting corollaries below.

Corollary 4.5 *Let G be a graph. Then the number of closed patterns that appear at least once in DB_G is even.*

Proof: *Suppose there are n closed patterns that appear at least once in DB_G, denoted as C_1, C_2, ..., C_n. As per Proposition 4.4, $occ(C_1)$, $occ(C_2)$, ..., $occ(C_n)$ are all closed patterns of DB_G. As per Proposition 4.1, $occ(C_i)$ is different from $occ(C_j)$ iff C_i is different from C_j. So every closed pattern can be paired with a distinct closed pattern by $occ(\cdot)$ in a bijective manner. Furthermore, as per Proposition 4.2, no closed pattern is paired with itself. This is possible only when the number n is even.* □

Corollary 4.6 *Let G be a graph. Then the number of closed patterns C, such that both C and $occ^{DB_G}(C)$ appear at least ms times in DB_G, is even.*

Proof: *As seen from the proof of Corollary 4.5, every closed pattern C of DB_G can be paired with $occ^{DB_G}(C)$, and the entire set of closed patterns can be partitioned into such pairs. So a pair of closed patterns C and $occ^{DB_G}(C)$ either satisfy or do not satisfy the condition that both C and $occ^{DB_G}(C)$ appear at least ms times in DB_G. Therefore, the number of closed patterns C, such that both C and $occ^{DB_G}(C)$ appear at least ms times in DB_G, is even.* □

Note that this corollary does not imply the number of closed patterns that appear at least ms times in DB_G is even. A counter example is given below.

Example 4.7 *Consider a DB_G given by the following matrix:*

	p_1	p_2	p_3	p_4	p_5
$\beta(p_1)$	0	1	1	0	0
$\beta(p_2)$	1	0	1	1	1
$\beta(p_3)$	1	1	0	1	1
$\beta(p_4)$	0	1	1	0	0
$\beta(p_5)$	0	1	1	0	0

We list its closed patterns, their support, and their $occ(\cdot)$ counterpart patterns below:

support of X	*close pattern X*	$Y = occ(X)$	*support of Y*
3	$\{p_2, p_3\}$	$\{p_1, p_4, p_5\}$	2
4	$\{p_2\}$	$\{p_1, p_3, p_4, p_5\}$	1
4	$\{p_3\}$	$\{p_1, p_2, p_4, p_5\}$	1

Suppose we take $ms = 3$. Then there are only 3 closed patterns—an odd number—that occur at least ms times, viz. $\{p_2, p_3\}$, $\{p_2\}$, and $\{p_3\}$.

Finally, we demonstrate our main result on the relationship with closed patterns and maximal complete bipartite subgraphs. In particular, we discover that every pair of a closed pattern C and its occurrence set $occ^{DB_G}(C)$ yields a distinct maximal complete bipartite subgraph of G.

Theorem 4.8 *Let G be an undirected graph without self-loop. Let C be a closed pattern of DB_G. Then the graph $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C)\rangle$ is a maximal complete bipartite subgraph of G.*

Proof: *By assumption, C is non-empty and C has a non-zero support in DB_G. Therefore, $occ(C)$ is non-empty. By Proposition 4.2, $C \cap occ^{DB_G}(C) = \{\}$. Furthermore, $\forall v \in occ(C)$, v is adjacent to every vertex of C. So, $C \times occ(C) \subseteq E^G$, and every edge of H connects a vertex of C and a vertex of $occ(C)$. Thus, H is a complete bipartite subgraph of G. By Proposition 3.1, we have $occ^{DB_G}(C) = \beta^G(C)$. By Proposition 3.2, $C = \beta^G(\beta^G(C))$. By Proposition 3.1, we derive $C = \beta^G(occ^{DB_G}(C))$. So H is maximal. This finishes the theorem.* □

Theorem 4.9 *Let G be an undirected graph without self-loop. Let graph $H = \langle V_1 \cup V_2, E\rangle$ be a maximal complete bipartite subgraph of G. Then, V_1 and V_2 are both a closed pattern of DB_G, $occ^{DB_G}(V_1) = V_2$ and $occ^{DB_G}(V_2) = V_1$.*
Proof: *Since H is a maximal complete bipartite subgraph of G, then $\beta(V_1) = V_2$ and $\beta(V_2) = V_1$. By Proposition 3.2, $CL(V_1) = \beta(\beta(V_1)) = \beta(V_2) = V_1$. So, V_1 is a closed pattern. Similarly, we can get V_2 is a closed pattern. By Proposition 3.1, $occ(V_1) = \beta(V_1) = V_2$ and $occ(V_2) = \beta(V_2) = V_1$, as required.* □

The above two theorems say that maximal complete bipartite subgraphs of G are all in the form of $H = \langle V_1 \cup V_2, E\rangle$, where V_1 and V_2 are both a closed pattern of DB_G. Also, for every closed pattern C of DB_G, the graph $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C)\rangle$ is a maximal complete bipartite subgraph of G. So, there is a one-to-one correspondence between maximal complete bipartite subgraphs and closed pattern pairs.

We can also derive a corollary linking support threshold of DB_G to the density of maximal complete bipartite subgraphs of G.

Corollary 4.10 *Let G be an undirected graph without self-loop. Then $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C)\rangle$ is a (m, n)-dense maximal complete bipartite subgraph of G iff C is a closed pattern such that C occurs at least m times in DB_G and $occ^{DB_G}(C)$ occur at least n times in DB_G.*

The corollary above has the following important implication.

Theorem 4.11 *Let G be an undirected graph without self-loop. Then $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C)\rangle$ is a (m, n)-dense maximal complete bipartite subgraph of G iff C is a closed pattern such that C occurs at least m times in DB_G and $|C| \geq n$.*

Proof: *Suppose $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C)\rangle$ is a (m, n)-dense maximal complete bipartite subgraph of G. By Theorem 4.9, $C = occ(occ(C))$. By definition of $occ(\cdot)$, $sup(occ(C)) = |occ(occ(C))| = |C|$. Substitute this into Corollary 4.10, we get H is a (m, n)-dense maximal complete bipartite subgraph of G iff C is a closed pattern such that C occurs at least m times in DB_G and $|C| \geq n$ as desired.* □

Theorems 4.8 and 4.9 show that algorithms for mining closed patterns can be used to extract maximal complete bipartite subgraphs of undirected graphs without self-loop. Such data mining algorithms are usually significantly more efficient at higher support threshold ms. Thus Theorem 4.11 suggests an important optimization for mining (m, n)-dense maximal complete bipartite subgraphs. To wit, assuming $m > n$, it suffices to mine closed patterns at support threshold $ms = m$, and then get the answer by filtering out those patterns of length less than n.

Table 1: Close patterns in a yeast protein interaction network.

support threshold	# of frequent close patterns	# of qualified close patterns	time in sec.
1	121314	121314	3.859
2	117895	114554	2.734
3	105854	95920	2.187
4	94781	80306	1.765
5	81708	60038	1.312
6	66429	36478	0.937
7	50506	15800	0.625
8	36223	3716	0.398
9	25147	406	0.281
10	17426	34	0.171
11	12402	2	0.109
12	9138	0	0.078

5 Experimental Results

We use an example to demonstrate the efficiency of listing all maximal complete bipartite subgraphs by using an algorithm for mining closed patterns. The graph is a protein interaction network with proteins as vertices and interactions as edges. As there are many physical protein interaction networks corresponding to different species, here we take the simplest and most comprehensive yeast physical and genetic interaction network (Breitkreutz et al., 2003) as an example. This graph consists of 4904 vertices and 17440 edges (after removing 185 self loops and 1413 redundant edges from the original 19038 interactions). Therefore, the adjacency matrix is a transactional database with 4904 items and 4904 transactions. On average, the number of items in a transaction is 3.56. That is, the average size of the neighborhood of a protein is 3.56.

We use FPclose* (Grahne & Zhu, 2003), a state-of-the-art algorithm for mining closed pattern, for enumerating the maximal complete bipartite subgraphs. Our machine is a PC with a CPU clock rate 3.2GHz and 2GB of memory. The results are reported in Table 1, where the second column shows the total number of **frequent** close patterns whose support level is at least the threshold number in the column one. The third column of this table shows the number of close patterns whose cardinality and support are both at least the support threshold; all such closed patterns are termed qualified closed patterns. Only these qualified closed patterns can be used to form maximal complete bipartite subgraphs $H = \langle V_1 \cup V_2, E\rangle$ such that both of $|V_1|$ and $|V_2|$ are at least a threshold. From the table, we can see:

- The number of all closed patterns (corresponding to those with the support threshold of 1) is even. Moreover, the number of qualified close patterns with cardinality no less than any support level is also even, as expected from Corollary 4.6.
- The algorithm is efficient—The algorithm takes less than 4 seconds to complete the program for all situations reported here. This indicates that enumerating all maximal complete bipartite subgraphs from a large graph can be efficiently solved by using algorithms for mining closed patterns.
- A so-called "many-few" property (Maslov & Sneppen, 2002) of protein interactions is observed again in our experiment results. The "many-few" property says that: a protein that interacts with lots of other proteins tends not to interact with another protein that interacts with lots of other proteins (Maslov & Sneppen, 2002). This is most clearly seen in Table 1 at the higher support thresholds. For example, at the support threshold 11, there are 12402 protein groups that have full interactions with at least 11 proteins. But there are only two groups that each contain at least 11 proteins and that have full mutual interaction.

6 Discussion and Conclusion

There are two recent research results related to our work. The problem of enumerating all maximal complete bipartite subgraphs (called maximal bipartite cliques there) from a *bipartite graph* has been investigated by (Makino & Uno, 2004). The difference is that our work is to enumerate all the subgraphs from any graphs (without self loops and undirected), but Makino and Uno's work is limited to enumerating from only bipartite graphs. So, our method is more general. Zaki (Zaki & Ogihara, 1998) observed that a transactional database DB can be represented by a bipartite graph H, and also a relation that closed patterns (wrongly stated as maximal patterns in (Zaki & Ogihara, 1998)) of DB one-to-one correspond to maximal complete bipartite subgraphs (called maximal bipartite clique there) of H. However, our work is to convert a graph G, including bipartite graphs, into a special transactional database DB_G, and then to discover all closed patterns DB_G for enumerating all maximal complete bipartite subgraphs of G. Furthermore, the occurrence set of a closed pattern of Zaki's work may not be a closed pattern, but that of ours is always a closed pattern.

Finally, let's summarize the results achieved in this paper. We have studied the problem of listing all maximal complete bipartite subgraphs from a graph. We proved that this problem is equivalent to the mining of all closed patterns from the adjacency matrix of this graph. Experimental results on a large protein interactions' data show that our method is efficient and the listing is fast.

References

Bastide, Y., Pasquier, N., Taouil, R., Stumme, G., & Lakhal, L. (2000). Mining minimal non-redundant association rules using frequent closed itemsets. *Computational Logic*, 972–986.

Breitkreutz, B. J., Stark, C., & Tyers, M. (2003). The grid: The general repository for interaction datasets. *Genome Biology*, *4*, R23.

Eppstein, D. (1994). Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, *51*, 207–211.

Goethals, B., & Zaki, M. J. (2003). Fimi'03: Workshop on frequent itemset mining implementations. *Third IEEE International Conference on Data Mining Workshop on Frequent Itemset Mining Implementations* (pp. 1–13).

Grahne, G., & Zhu, J. (2003). Efficiently using prefix-trees in mining frequent itemsets. *Proceedings of FIMI'03: Workshop on Frequent Itemset Mining Implementations*.

Gross, J. L., & Yellen, J. (2004). *Handbook of graph theory*. CRC Press.

Makino, K., & Uno, T. (2004). New algorithms for enumerating all maximal cliques. *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT 2004)* (pp. 260–272). Springer-Verlag.

Maslov, S., & Sneppen, K. (2002). Specificity and stability in topology of protein networks. *Science*, *296*, 910–913.

Murata, T. (2004). Discovery of user communities from web audience measurement data. *Proceedings of The 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)* (pp. 673–676).

Pan, F., Cong, G., Tung, A. K. H., Yang, J., & Zaki, M. J. (2003). CARPENTER: Finding closed patterns in long biological datasets. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 637–642).

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. *Proceedings of the 7th International Conference on Database Theory (ICDT)* (pp. 398–416).

Pei, J., Han, J., & Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. *Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (pp. 21–30).

Reiss, D. J., & Schwikowski, B. (2004). Predicting protein-peptide interactions via a network-based motif sampler. *Bioinformatics (ISMB 2004 Proceedings), 20 (suppl.)*, i274–i282.

Tong, A. H., Drees, B., Nardelli, G., Bader, G. D., Brannetti, B., Castagnoli, L., Evangelista, M., Ferracuti, S., Nelson, B., Paoluzi, S., Quondam, M., Zucconi, A., Hogue, C. W., Fields, S., Boone, C., & Cesareni, G. (2002). A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science, 295*, 321–324.

Uno, T., Kiyomi, M., & Arimura, H. (2004). Lcm ver.2: Efficient mining algorithms for frequent/closed/maximal itemsets. *IEEE ICDM'04 Workshop FIMI'04 (International Conference on Data Mining, Frequent Itemset Mining Implementations)*.

Wang, J., Han, J., & Pei, J. (2003). CLOSET+: Searching for the best strategies for mining frequent closed itemsets. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), Washington, DC, USA* (pp. 236–245).

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An efficient algorithm for closed itemset mining. *Proceedings of the Second SIAM International Conference on Data Mining.*

Zaki, M. J., & Ogihara, M. (1998). Theoretical foundations of association rules. *Proc. 3rd SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*.