Mining networks with shared items

Conference Paper · January 2010

DOI: 10.1145/1871437.1871703 · Source: DBLP

Jun Sese
Tokyo Institute of Technology

Mining Networks with Shared Items

Jun Sese
Dept. of Computer Sciences,
Ochanomizu Univ.
sesejun@is.ocha.ac.jp

Mio Seki
Dept. of Computer Sciences,
Ochanomizu Univ.
seki@sel.is.ocha.ac.jp

Mutsumi Fukuzaki
Dept. of Computer Sciences,
Ochanomizu Univ.
fukuzaki@sel.is.ocha.ac.jp

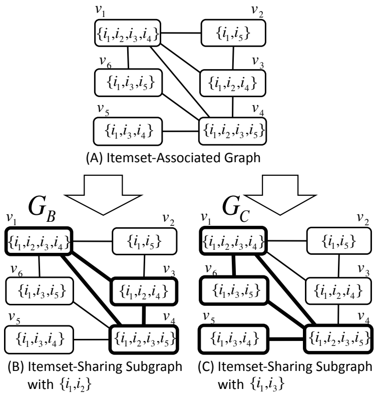
ABSTRACT

Recent advances in data processing have enabled the generation of large and complex graphs. Many researchers have developed techniques to investigate informative structures within these graphs. However, the vertices and edges of most real-world graphs are associated with its features, and only a few studies have considered their combination. In this paper, we specifically examine a large graph in which each vertex has associated items. From the graph, we extract subgraphs with common itemsets, which we call *itemset-sharing subgraphs (ISSes)*. The problem has various potential applications such as the detection of gene networks affected by drugs or the findings of popular research areas of contributing researchers. We propose an efficient algorithm to enumerate ISSes in large graphs. This algorithm enumerates ISSes with two efficient data structures: a *DFS itemset tree* and a *visited itemset table*. In practice, the combination of these two structures enables us to compute optimal solutions efficiently. We demonstrate the efficiency of our algorithm in mining ISSes from synthetic graphs with more than one million edges. We also present experiments performed using two real biological networks and a citation network. The experiments show that our algorithm can find interesting patterns in real datasets.

Categories and Subject Descriptors: H.2.8 [Database Applications]: data mining
General Terms: Algorithms, Performance
Keywords: Large Graph, Itemset, Gene Network, Citation Network, Social Network

1. INTRODUCTION

Large graphs can represent real-world relationships, and there has been considerable interest in graph analysis in various domains. Researchers have studied the common properties of network structures with various degrees of vertices [2], and have developed techniques to find dense subgraphs within graphs efficiently [10]. At the same time, data mining researchers have extracted frequent subgraphs from graph databases [4, 13].



(A) Itemset-Associated Graph

(B) Itemset-Sharing Subgraph with $\{i_1,i_2\}$

(C) Itemset-Sharing Subgraph with $\{i_1,i_3\}$

Figure 1: An Example of Itemset-Associated Graph and Itemset-Sharing Subgraphs.

Despite these studies of network structures, most real-world networks are related to complex features, which are informative for characterizing portions of the large graph. However, only a few studies have considered combinatorial mining of the two types of data structures [11].

In this paper, we specifically examine a novel graph model, which is an undirected graph whose vertices have itemsets. This graph appears in various situations: a social network in which each vertex represents a person and contains a record of that person's purchased products; a gene network in which each vertex represents a gene and is associated with that gene's activated conditions; and a citation network in which each vertex is a paper and is labeled with its respective authors. An example of the graph is presented in Figure 1(A).

The graph shows that one pattern of interest is a subgraph containing itemsets of vertices that have common items. For example, the subgraph of a social network that shares a set of purchased products would indicate that the friends bought the items on the basis of word-of-mouth communication. We often purchase the same product purchased by our friends when they tell or show us good points about the product. An additional example is a segment of a metabolic pathway, a gene network, correlated with the activated conditions of genes might indicate the working time of the segment. Knowledge of an activated network and its conditions might be helpful for making new drugs.

These applications have prompted us to consider a novel connected subgraph that contains vertices that share an itemset, called an *itemset-sharing subgraph (ISS)*. However, it is difficult to enumerate such subgraphs because the num-

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

ber of subgraphs increases exponentially according to the size of a given graph. To overcome this difficulty, we designed an algorithm called COmmon Itemset Network mining (*COIN*), which computes the optimal solution and contains the following two data structures: *depth-first search (DFS) itemset tree* and *visited itemset table*. By combining the two structures, we can reduce the search space dramatically. We demonstrate the scalability of our algorithm with respect to the amount of data by using synthetic data, and we demonstrate the algorithm's application to a real biological network dataset and a citation network.

2. PROBLEM DEFINITION

In this section, we formally define the problem of finding the maximum ISS.

Let G be an undirected, unlabelled, unweighted graph where each vertex has a set of items (an itemset). We define this graph as the *itemset-associated graph (IA graph)*. Let $V(G)$, $E(G)$, $\mathcal{I}(G)$, and $I(v)$ respectively signify a set of vertices in G, a set of edges in G, a set of itemsets on vertices in G, and an itemset on $v \in V(G)$. For this description, $|G| = |E(G)|$ is the size of graph G. Figure 1(A) portrays an example of the IA graph. Each vertex in G has an itemset. Next, we define a graph having common items.

Definition 1. *(Subgraph) Let G be an IA graphs. We define $I(G)$ as $\bigcap_{v \in V(G)} I(v)$. We call $I(G)$ as a shared itemset of G. We define G' a subgraph of G when $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$ and $I(G') \supseteq I(G)$. We describe $G' \subseteq G$ when G' is a subgraph of G.*

With this definition of subgraph in terms of an IA graph, we define an important feature of a subgraph.

Definition 2. *(Itemset-Sharing Subgraph (ISS)) Let G' be a connected subgraph of the IA graph G. G' is also an IA graph. When $I(G') \neq \phi$, we say that G' is an itemset-sharing subgraph (ISS) with $I(G')$.*

We present two examples of ISSes in Figures 1(B) and (C), which are indicated by the bold lines.

To introduce the formal definition of ISS discovery, we define a *connected graph* in terms of an IA graph.

Definition 3. *(Connected Graph) Let G be an IA graph, and G_1 and G_2 be the subgraphs of G and ISSes. We say that G_1 and G_2 are mutually connected when $E(G_1) \cap E(G_2) \neq \phi$ and $|I(G_1) \cap I(G_2)| = \min\{|I(G_1)|, |I(G_2)|\}$.*

We only consider the graphs to be connected when the inclusion relation between the itemsets of the two graphs exists. By this definition, G_B and G_C shown in Figures 1(B) and (C) are disconnected.

If connected graphs exist, we are interested in the largest one. The following definitions of the largest ISS enable us to select the ISSes of interest.

Definition 4. *(Largest ISS) Given an IA graph G. Let G' be an ISS with I. When no edge $(v_1, v_2) \in G$, where $v_1 \in G'$ satisfies $I(V(G) \cup \{v_2\}) = I$, G' is called maximal ISS with I. We call the maximal ISS whose size is the largest as the largest ISS, and the N largest maximal ISSes as the N largest ISSes.*

With this definition, we formalize the problem of finding the N largest ISSes.

Definition 5. *(Finding the N Largest ISSes) Given an IA graph G, user-specified values N and θ_I, we compute the N largest ISSes G' such that $|I(G')| \geq \theta_I$.*

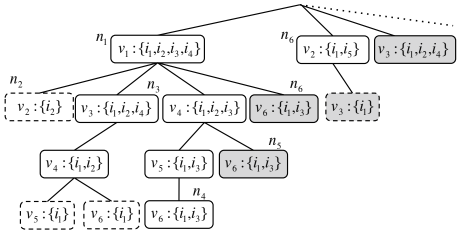


Figure 2: A DFS Itemset Tree

Unfortunately, it is difficult to compute the N largest subgraphs because the number of subgraphs increase exponentially with the size of a graph. Therefore, to overcome this difficulty, we introduce two data structures: a DFS itemset tree and a visited itemset table.

3. MINING ITEMSET-ASSOCIATED GRAPHS

To enumerate all of the subgraphs, two different approaches exist: April-like generations (breadth-first search) [4]and a pattern-growth approach (DFS) [13]. We designed a novel algorithm COIN using the pattern-growth approach.

DFS Itemset Tree.

We use a DFS tree for enumerating ISSes $\mathcal{G}$ where $|G| \geq \theta_S$ and $|I(G)| \geq \theta_I$ for $G \in \mathcal{G}$. Each node of the tree contains a vertex and an itemset related to the path from the root to the node. We denote this tree as a *DFS itemset tree*. We do not need to maintain edges in the DFS itemset tree because $I(G)$ can be computed from the vertices and their itemsets.

The generation of the subgraphs themselves is considered to be a simplified version of the DFS lexicographic order used in the gSpan algorithm [13], and hence, this DFS itemset tree can avoid duplicate generation of identical graphs. Fig. 2 shows the DFS itemset tree for the IA graph in Fig. 1(A). Each node in the DFS itemset tree contains a vertex and an itemset. The vertices included in the path from the root to the tree node represent the vertices of the subgraph.

As a result of the following monotonic property of ISSes regarding the itemset size, we can prune subtrees in the DFS itemset tree, which dramatically reduces the search space.

Property 1. *Given an IA graph G. Let G_p and G_c be subgraphs of G. When $V(G_p) \subset V(G_c)$, $I(G_p) \supseteq I(G_c)$.*

As a result of Property 1, the sizes of the itemsets of the DFS itemset tree have the following monotonic property: if the common itemset size in a node n of the DFS itemset tree is less than θ_I, the common itemset size in any descendant node n' of n is less than θ_I.

Visited Itemset Table.

In this section, we introduce a visited itemset table, which enables us to avoid visiting connected subgraphs. Each record in this table contains a vertex and itemsets. This table gives clues about whether the descendants of a node in the DFS itemset tree become disconnected in relation to the already generated subgraphs. We can prune the subtree of the node immediately if the subgraphs are inseparable.

To execute such pruning, we introduce a relation between two ISSes.

Lemma 1. *Let v be a vertex and $I \subseteq I(v)$. Only one ISS associated with I contains v.*

Theorem 1. *Let n_1 and n_2 be the nodes of the DFS itemset tree and n_1 be generated before n_2. If both n_1 and n_2 contain v and $I(n_1) \supseteq I(n_2)$, no maximal ISS exists in a descendant of n_2.*

Table 1: A Visited Itemset Table.

Vertex v	$\mathcal{P}(v)$
v_1	$\{i_1, i_2, i_3, i_4\}$
v_3	$\{i_1, i_2, i_3\}$
v_4	$\{i_1, i_2\}, \{i_1, i_2, i_3\}$
v_5	$\{i_1, i_3\}$
v_6	$\{i_1, i_3\}$

Table 2: Parameters.

Name	Description	Default
$\|V\|$	The number of vertices	1,000
$\|E\|$	The number of edges	5,000
N	The number of items	100
$\|T\|$	The average size of the itemsets in each vertex	20
$\|I\|$	The itemset size shared by the largest ISS	10
$\|L\|$	The size of the largest ISS	20
θ_I	The minimum common itemset size	$\|I\|$

Table 3: The largest 5 networks from the DBLP dataset.

No.	Authors	# of refs.	# of papers
1	Miron Livny, Michael J. Carey	48	26
2	Philip S. Yu, Ming-Syan Chen	43	16
3	Philip S. Yu, Daniel M. Dias	36	23
4	Victor Vianum, Serge Abiteboul	33	24
4	Divyakant Agrawal, Amr El Abbadi	33	20

This property implies that if we visit an already visited node v and the shared itemset of the current path is the same or a subset of an itemset when we previously visited on v, we can prune the subtree rooted by the current node in the DFS itemset tree. Therefore, this property enables us to avoid unnecessary subgraph exploration.

Theorem 1 prompts us to make the following hash table for the efficient pruning of subgraphs.

DEFINITION 6. *(Visited Itemset Table) We denote a set of itemsets associated with a vertex v as $\mathcal{P}(v)$. We add I to $\mathcal{P}(v)$ when we visit a node $n = (v, I)$ of the DFS itemset tree. We define the hash table as a visited itemset table.*

While constructing a DFS itemset tree, we can build a visited itemset table. Table 1 shows the contents of $\mathcal{P}$ when we visit n_4 in Figure 2.

However, two problems might occur. First, it might be too costly to check an inclusion relation in a visited itemset table. Second, in a real network, this pruning technique might not work well. With regard to these problems, in Section 4, we show that our algorithm is sufficiently fast to solve problems dealing with synthetic and real data. With the DFS itemset tree and visited itemset table, we designed a novel algorithm to quickly discover the largest common pattern graph.

4. EXPERIMENTAL RESULTS

In this section, we describe the performance of the COIN algorithm. All experiments were performed on a 2.2 GHz AMD Opteron machine with 1GB main memory, running on Linux 2.6. The COIN algorithm was implemented using Java 5. First, we experimented using synthetic data and then we showed the results obtained using a real biological dataset and a DBLP dataset.

Synthetic Data. We generated a synthetic IA graph to evaluate the performance of the COIN algorithm over a large range of data characteristics. Their graph structures match those of a random graph.

Performance Results. We introduced the visited itemset table, but we must consider the fact that the scanning cost of the table might be high. We therefore examined the effectiveness of the table using synthetic data. Figure 3 (A) shows the execution times obtained with and without the use of the visited itemset table, which is indicated by "COIN" and "w/o Visited Itemset Table", respectively. We changed the degrees to check the dependency of the network complexity. An increase in the average degree can result in numerous graphs that do not yield the largest ISS. The execution time increases exponentially, and almost quadratically, both with and without the use of the visited itemset table. Note that the y axis is on a log scale. From this figure, COIN is shown to be about an order of magnitude faster than the algorithm without the visited itemset table.

Figure 3(B) reports the execution time of COIN as a function of the number of edges (the number of edges is the average degree × number of nodes). Although the number of subgraphs increases exponentially with the number of edges, this figure shows that the execution times increase conservatively. As presented in this figure, COIN can reach a solution quickly even when the number of edges exceeds 200K (# of nodes = 10K and the average degree = 20).

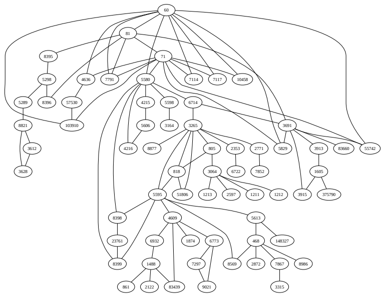


Figure 4: The largest ISS of the metabolic pathway associated with a protein kinase.

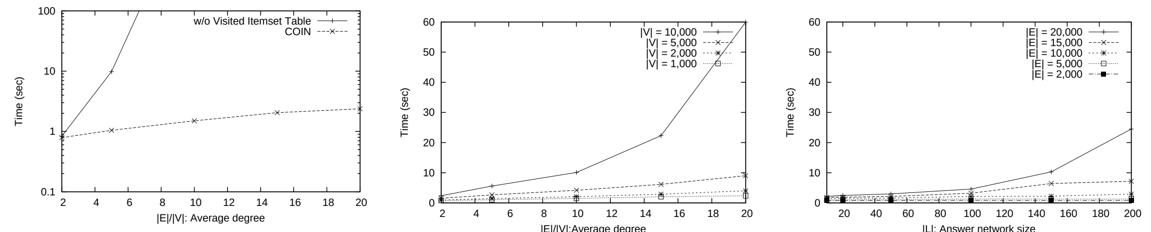
Figure 3(C) shows the execution time as a function of the complexity or the pattern size of an answer network. An increase in the size of an answer network might increase the cost of network exploration exponentially because the number of subgraphs increases more than exponentially. However, our results show that the execution time increases almost quadratically. This result indicates that our pruning technique works efficiently.

Real Biological Data.

We investigated the effectiveness and usefulness of finding the largest ISS by examining a biological dataset with a metabolic pathway network in humans [5].

The metabolic pathway network has 1,871 nodes and 10,871 edges; the nodes and edges are genes and chemical interactions, respectively. Fro the itemset at each vertex, we converted the human microarray data under 28 different conditions [9] into a Boolean set using a threshold of 1.5.

Figure 4 shows that the largest ISS in the metabolic pathway network, which has a scale-free propertyand a hierarchical network modularity. The ISS was calculated under $t = 0.25$ and $\theta_I = 3$. The network has 68 vertices and 94 edges, and their shared itemset is {HRG_U0126 5min, 15min, 30min}. The gene expression data utilized contains time course data at 5, 10, 15, 30, 45, 90 and 180 min from four different stimuli. The shared itemset contains similar times for the same stimuli; hence, our method successfully extracted the ISS associated with time course. In this figure, each node signifies a gene (the number is the NCBI gene ID), and each edge represents the chemical reaction in a cell. The result shows a hierarchical network in the pathway. This network contains the protein kinases, which hierarchically controls information within a cell. In contrast to clique finding methods, this result shows that our



(A) Effect of the visited itemset table according to the degree of nodes

(B) Scalability with number of edges

(C) Scalability for answer network size

Figure 3: Performance results.

method can find the sparse subgraphs sharing items; hence our method can extract the modules of signal transduction cascade that form a hierarchical structure.

DBLP Dataset.

We investigated the usefulness of the largest ISS by using the DBLP dataset [6] consisting of a snapshot as of April 12, 2006. The DBLP network has 22,178 papers (vertices), 112,304 references (edges) and 16,638 authors (items). All of the papers had at least one author and one reference. The average number of authors over all the papers is 2.29. The parameters for COIN were set to $\theta_I = 2$ and $\theta_S = 10$. The computation time was 7.9s on the same machine and environment as the synthetic dataset.

Table 3 shows the top 5 largest network results extracted by COIN. All of the researchers in the Top 5 networks are famous within the database and data mining community; hence, our method can effectively extract the important networks. All of the papers in this network contains at most 33 papers, and the support value of the number of itemsets was $33/22,178 \approx 0.14\%$. It is difficult to find association rules in such a very low support network.

In this experimental result, we used author names as items. If the authors of every paper were replaced by keywords about the papers, COIN also can extract the reference network that has members that share topics.

5. RELATED WORK

One might think that graph clustering methods [7] provide solutions to the ISS enumeration. Although some methods use the edge weight, our common pattern graphs might share one edge between two different graphs such as the edge (v_1, v_4) in Figures 1(B) and (C). Therefore, we require a method that specifically addresses the common itemset in subgraph.

Upon the ISS enumeration, frequent pattern mining [1, 3] might provide the first step finding ISSes because the frequent itemset mining constructs a large ISS. However, since support of the largest ISS might be very low, the development of an efficient algorithm is required to find ISSes.

MATISSE [12] and CoPaM [8] study the combinatorial mining of networks with feature vectors. Both methods find dense subgraphs with vertices that have similar features. However, we are not concerned with the density of the subgraphs, and our method can find the *sparse* hub network shown in Figure 4.

6. CONCLUDING REMARKS

In this paper, we introduced a new graph structure, namely, an undirected relational graph in which the nodes contain a set of items. We also proposed a novel graph mining problem called *itemset-sharing subgraph (ISS) problem*, which deals with finding the largest network in which vertices share more items than a threshold.

However, it is difficult to solve this problem because the number of subgraphs increases exponentially with the size of the given graph. To solve the problem efficiently, we introduced a novel algorithm *COIN*, which generates subgraphs using the pattern-growth approach along with two new data structure: the DFS itemset tree and the visited itemset table. The combination of these two datasets enabled us to dramatically reduce the search space. Our demonstration with synthetic data shows that our algorithm is effective even in a large and complicated network. Furthermore, the application of our model to a biological and an author network proved that our algorithm extracts informative subgraphs.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB 1994*, pages 487–499, 1994.

[2] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–6, Jun 2002.

[3] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD '00*, pages 1–12, 2000.

[4] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD '00*, 2000.

[5] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nuc. Acids Res.*, 28(1):27–30, 2000.

[6] Knowledge Discovery Laboratory, University of Massachusetts Amherst. The Proximity DBLP database. http://kdl.cs.umass.edu/data/dblp/dblp-info.html.

[7] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. In *ICML '05*, pages 457–464, 2005.

[8] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM '09*, 2009.

[9] T. Nagashima et al. Quantitative transcriptional control of erbb receptor signaling undergoes graded to biphasic response for cell differentiation. *J. Biol. Chem.*, 282:4045–4056, 2007.

[10] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167, 2003.

[11] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numericalvectors with a modular network. In *KDD '07*, pages 647–656, 2007.

[12] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high throughput data. *BMC Systems Biology*, 1, 2007.

[13] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM '02*, page 721, 2002.