Mining Top-K Graph Patterns that Jointly Maximize Some Significance Measure

Yong Liu
Harbin Institute of Technology, Harbin, Country
Email: liuyong123456@hit.edu.cn

Jianzhong Li, Jinghua Zhu, Hong Gao
Harbin Institute of Technology, Harbin, Country

Abstract—**Most of graph pattern mining algorithms focus on finding frequent subgraphs and its compact representations, such as closed frequent subgraphs and maximal frequent subgraphs. However, little attention has been paid to mining graph patterns with user-specified significance measure. In this paper, we study a new problem of mining top-k graph patterns that jointly maximize some significance measure from graph databases. Exploiting entropy and information gain, we give two problem formulations, EM and IGM. We first prove them to be NP-hard and then propose two efficient algorithms, PP-TopK and DM-TopK, to solve them. PP-TopK greedily selects top-k graph patterns among frequent subgraphs. DM-TopK integrates the pruning techniques into the mining framework, and directly mines top-k graph patterns from graph databases. Empirical results demonstrate the quality of our top-k graph patterns, and validate the efficiency and scalability of our algorithms.**

Index Terms—**pattern mining, graph database, frequent subgraph, top-k.**

I. Introduction

As a general data structure, graph can be used to model complicated relations among data objects. It has been prevalently used in a wide range of application domains, such as protein interaction graphs in biology [1], chemical compound structures in chemistry [2], social networks in social science [3] and web graphs [4]. As witnessed in the core tasks of many applications, including graph search [5, 6] and classification [7], graph patterns could help build powerful, yet intuitive models for better managing and understanding complex structures. Their usage, therefore, is far beyond traditional exercises, such as association rules. With the increasing popularity of graph databases in various applications, discovering useful graph patterns from graph databases emerges as one of the most important mining problems.

In literature, there have been a number of studies on mining interesting patterns from graph databases. Most of them focus on finding frequent subgraphs [8-13] and its compact representations, such as closed frequent subgraphs [14] and maximal frequent subgraphs [15,16]. However, little attention has been paid to mining graph patterns based on a significance measure specified by users. Furthermore, users may need different significance measures in different applications. Essentially, support is also a specific significance measure, which is widely used in mining frequent subgraphs.

Given a significance measure *M*, a natural idea is to rank discovered graph patterns or all graph patterns in the database, and output top-k patterns as the traditional top-k query/mining method. However, the traditional top-k method assumes that patterns are independent from each other, which unfortunately is not the case. Fig. 1 in Section V shows the traditional top-5 graph patterns in a database of chemical compounds when information gain is used as a significance measure. We find that these five graph patterns overlap substantially with respect to structure, and thus they are not independent from each other. If users obtain one of them, the other graph patterns become insignificant for users.

To overcome the shortcomings of the traditional top-k method, in this paper, we consider the joint significance for a set of graph patterns instead of the single significance for each pattern. Specially, given a graph database *D* and a significance measure *M*, the problem studied in this paper is to discover a set *T* consisting of *k* graph patterns from *D* such that *M*(*T*) is maximized. The joint significance measure will present a set of significant yet distinct graph patterns to users. For example, Fig. 2 in Section V shows the top-5 graph patterns generated based on the joint information gain. They are different from each other with respect to structure, which is desirable for users in many applications.

In this paper, we study the problem of mining top-k graph patterns jointly maximizing some significance measure from a graph database. In order to describe our method clearly, we first use two concepts in information theory, entropy and information gain, to give two concrete problem formulations, EM and IGM. We then prove that EM and IGM are NP-hard. We propose two mining method, PP-TopK and DM-TopK, for EM and IGM. PP-TopK is a post-processing method which first mines frequent graph patterns from graph databases, and then selects top-k graph patterns from frequent patterns. DM-TopK integrates effective pruning techniques designed for some significance measure into the framework of frequent subgraph mining algorithms, and directly mines top-k graph patterns jointly maximizing some significance measure from graph databases.

doi:10.4304/jcp.5.4.565-572

Furthermore, both PP-TopK and DM-TopK can provide an offline approximation bound for EM, and provide an online approximation bound for IGM.

Extensive experiments show that our top-k methods, PP-TopK and DM-TopK can generate better results than the traditional top-k method in terms of classification accuracy. Compared with PP-TopK, DM-TopK has a higher efficiency, especially when the minimum support is low. We summarize the contributions of this paper as follows.

- We propose a novel problem of mining top-k graph patterns that jointly maximize some significance measure from graph databases.
- We give two problem formulations, EM and IGM, using entropy and information gain, and prove that they are NP-hard.
- We propose two efficient algorithms, PP-TopK and DM-TopK, for this new problem. Furthermore, we prove that both PP-TopK and DM-TopK can provide an offline approximation bound for EM and an online approximation bound for IGM.
- Extensive study has been conducted to demonstrate the quality of top-k graph patterns and validate the algorithm's efficiency and scalability.

The rest of the paper is organized as follows. We introduce some background knowledge on frequent subgraph mining and information theory in Section II. We give two problem formulations, EM and IGM, using entropy and information gain, and prove that they are NP-hard in Section III. Two efficient mining algorithms, PP-TopK and DM-Topk, are proposed in Section IV. Then, we evaluate the performance of PP-TopK and DM-Topk and demonstrate the quality of mining results in Section V. Finally, we review related work in Section VI and conclude the paper in Section VII.

II. Preliminaries

In this section, we introduce some preliminary concepts and notations on frequent subgraph mining and information theory.

Definition 1 (Labeled Graph). *A labeled graph G is a 4-tuple G = (V, E, L, l), where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, L is the set of labels and l: V□E →L is a labeling function that maps each vertex or edge to a label in L.*

Definition 2 (Subgraph Isomorphism). *A labeled graph G = (V, E, L, l) is subgraph isomorphic to a labeled graph G' = (V', E', L', l') iff there exists an injective function f: V→V', such that (1) $\forall u \in V$, l(u) = l'(f(u)) and (2) $\forall (u, v) \in E$, $(f(u), f(v)) \in E'$ and l(u, v) = l'(f(u), f(v)).*

If there exists a subgraph isomorphism from G to G', G is called a subgraph of G' and G' is called a supergraph of G, denoted by $G \subseteq G'$. If $G \subseteq G'$ and $G \neq G'$, G is called a proper subgraph of G', denoted by $G \subset G'$. The subgraph isomorphism testing is an NP-complete problem [17]. If $G \subseteq G'$, we call that G' contains G.

A graph database D consists of a set of labeled graphs whose cardinality is denoted by |D|. The **support set** of a graph pattern p in D is the set of all graphs in D that are supergraphs of p. The **support** of a graph pattern p in D, denoted by $supp_D(p)$, is defined as the number of graphs in D that are supergraphs of p. When the context is clear, we use supp(p) to denote $supp_D(p)$. The support measure is **anti-monotone**, i.e., if $p_1 \subseteq p_2$, then $supp(p_1) \geq supp(p_2)$. A graph pattern p is **frequent** in D if its support in D is no less than a user-specified threshold *min_sup*. A graph pattern p is **closed** in D if there exists no proper supergraph of p that has the same support as p. The set of **frequent graph patterns** is denoted by FS. The set of **closed frequent graph patterns** is denoted by CS, i.e., $CS = \{p | p \in FS$ and $\neg\exists\, p' \in FS$ such that $p \subset p'$ and $supp(p) = supp(p')\}$.

If not explicitly stated, the notation "graph" means an *undirected labeled connected graph* by default in the rest of this paper. Since our proposed significant measures exploit some concepts of information entropy [18], we review them briefly.

Definition 3 (entropy). *The entropy of a random variable x, denoted as H(x), is defined as* $H(x) = -\sum_{v_x \in dom(x)} (p(v_x) log(p(v_x)))$, *where dom(x) is the domain of x.*

Definition 4 (conditional entropy). *The conditional entropy of a random variable y given another variable x, denoted as H(y|x), is defined as* $H(y|x) = -\sum_{v_x \in dom(x)} \sum_{v_y \in dom(y)} (p(v_x, v_y) log(p(v_y|v_x)))$.

Definition 5 (joint entropy). *The joint entropy of two random variables x and y, H(x,y), is defined as* $H(x,y) = -\sum_{v_x \in dom(x)} \sum_{v_y \in dom(y)} (p(v_x, v_y) log(p(v_x, v_y)))$.

III. Problem Statement

In this section, we first discuss significance measures for a set of graph patterns, and propose the formal problem formulation. Then, we show that the proposed problem is NP-hard.

There are lots of studies on the significance measure for a single pattern. For example, Chi-square test, Pearson correlation, *etc.* can measure the statistical significance of a single pattern. In data mining and machine learning, discriminative measures such as information gain and cross entropy can be used to measure the discriminative power of a single pattern. Tan et al. [19] summarized twenty-one significance measures.

Although these significance measures are still valid for a single graph pattern, most of them cannot be used to measure the significance of a set of (graph) patterns. In this paper, we hope to find a measure that can show the combined significance for a set of graph patterns. Before giving the concrete significance measure, we first formulate the general problem studied in this paper as follows.

We denote by S the full set of available graph patterns, which corresponds to the frequent subgraphs generated by some frequent subgraph mining algorithm in this paper. Let M(T) be a significance measure for a set of graph patterns T. We want to discover a subset T^* of S such that

$$T^* = argmax_{T \subset S, |T|=k} M(T) \qquad (1)$$

If we first obtain all candidate graph patterns, then the problem defined above is clearly an combinatorial optimization problem. In this paper, our goal is to design a general mining method applicable to a wide range of significance measures. In order to describe our method clearly, we use information theory concepts to give the concrete problem formulation as follows.

Definition 6 (Entropy-Based Significance). *Given a graph database D (or a set of frequent subgraphs mined from D), the problem of maximizing entropy-based significance (**EM**) is to find a set of (frequent) subgraphs T such that H(T) is maximized, where H(T) is the joint entropy for graph patterns in T.*

Definition 7 (Information Gain-Based Significance). *Given a graph database D (or a set of frequent subgraphs mined from D). Assume that each graph in D has a class label, and C denotes the set of all class labels. The problem of maximizing information gain-based significance (**IGM**) is to find a set of (frequent) subgraphs T such that IG(T)= H(C) − H(C|T) is maximized, where H(C|T) is the entropy conditioned on T, and IG(T) is the information gain given T.*

The entropy-based significance is often used to measure uncertainly in an unsupervised setting, whereas the information gain-based significance is often used to select discriminative features in a supervised setting. In the above formulation, we consider a subgraph as a random variable. If a subgraph *p* occurs in a graph in the database, then the value of *p* is 1, otherwise 0.

Generally, when a concrete significant measure is given, the problem defined in Equation 1 is an NP-hard problem. In the following, we show that the problem of the information gain-based formulation (Definition 7) is NP-hard. The NP-hardness of the problem of the entropy-based formulation (Definition 6) can be proved similarly.

Theorem 1. *The problem of maximizing information gain-based significance (IGM) is NP-hard.*

Proof. We can easily prove this theorem by reduction from the MAX-COVER problem.□

IV. Efficient Mining of Top-K Graph Patterns

In this section, we illustrate how to efficiently mine top-k graph patterns that jointly maximize some significance measure.

A. Post-Processing Method

We have shown that both EM and IGM are NP-hard problems. Thus, we have to use approximation or heuristic algorithms to solve them. In this subsection, we discover top-k graph patterns jointly maximizing entropy and information gain from the set of frequent subgraphs *S*.

Since it is difficult to find the optimal solution, we adopt a well-known greedy algorithm to solve EM and IGM. The algorithm incrementally selects patterns from *S* with an estimated benefit *b*. A pattern is selected if it has the maximum benefit among the remaining patterns. That is, given a set of selected patterns *T*, the benefit of a pattern $p \in S-T$ is:

$$b(p) = \begin{cases} H(T,p) - H(T), & \text{for EM,} \\ H(C|T) - H(C|T,p) & \text{for IGM.} \end{cases} \quad (2)$$

Based on the greedy rule above, we devise a greedy post-processing algorithm, PP-TopK, as shown in Algorithm 1. At beginning, the result set *T* is empty. The algorithm picks the most significant pattern and inserts it into *T*. When $|T| < k$, we will compute benefit $b(p)$ for every remaining graph pattern $p \in S-T$, and select the pattern with the maximum benefit. After a pattern is inserted into *T*, it remains in *T*. When $|T| = k$, PP-TopK terminates.

Algorithm 1: **PP-TopK**

Input: (1) A graph Dataset *D*, (2) A minimum support *min_sup*, and (3) A significance measure *M*, and (4) Number of output patterns *k*.

Output: The set of *k* graph patterns *T* that jointly maximize *M*

```
Mine all frequent subgraphs $S$ from $D$ with $min\_sup$;
Select a graph pattern $p$ that maximizes $M(p)$;
$T = \{p\}$;
WHILE ($|T| < k$) do
    Find a graph pattern $p^*$ from patterns in $S - T$ such that $b(p^*)$ is maximized;
    $T = T \cup \{ p^* \}$;
Output $T$;
```

If a significance measure satisfies the submodular property (A measure function *M* is said to be submodular on a set *S* if for $T \subset T' \subseteq S$ and $p \in S$, $M(T \cup \{p\}) - M(T) \geq M(T' \cup \{p\}) - M(T')$, the greedy strategy above will yield a near-optimal solution [20].

We find that the entropy-based measure satisfies the submodular property, thus we can obtain the following approximation bound for EM.

Theorem 2. *Let the set of graph patterns selected by Algorithm 1 be T, the set of optimal k graph patterns that maximize the joint entropy H be T*, then* $H(T) \geq (1-1/e) \times H(T^*)$.

The information gain-based measure doesn't satisfy the submodular property, thus we cannot obtain any offline approximation bound for IGM. However, since *H*(*C*) is the possible maximum value of *IG*(*C*|*T*) for any set of graph patterns *T*, we can easily obtain an online approximation bound for IGM.

Theorem 3. *Let the set of graph patterns selected by Algorithm 1 be T, then the approximation bound for* IGM *is guaranteed to be at most H(C)/IG(T).*

B. Direct Mining

In this subsection, we design effective pruning techniques and integrate them into the framework of frequent subgraph mining algorithms to directly mine top-k graph patterns jointly maximizing entropy and information gain from graph databases.

All of frequent subgraph mining algorithms exploits the well-known anti-monotone property of support to prune the search space of frequent graph. That is, the support of a subgraph *p* is an upper bound for the supports of its supergraphs. However, the significance measures presented in this paper are not anti-monotone. We cannot apply the significance measures like support.

Fortunately, given the significance measure value of a subgraph *p*, we can derive an upper bound for the significance measure value of its supergraphs, thereby allowing us to effectively prune the search space for graph patterns.

We still adopt the framework of greedy selection to mine the top-k graph patterns. That is, given the current set of selected graph patterns *T*, we then select a graph pattern *p* such that $M(T\cup\{p\})$ is maximized, where *M* is a measure to be optimized.

In the following, we introduce the pruning techniques in two cases: (1) when the result set is empty, we mine the first graph pattern; (2) when some graph patterns have been selected, we mine the next graph pattern.

We first study how to effectively prune the search space when mining the first graph pattern. Theorem 4 gives the pruning condition for the entropy measure, and Theorem 5 gives the pruning condition for the information gain measure.

Theorem 4. *Let D be a graph database, and q be a supergraph of p. If* $supp_D(p) \le 1/2$*, then* $H(q) \le H(p)$.

Theorem 5. *Let D=PD+ND, where PD and ND are positive and negative graph databases respectively, q be a supergraph of p, x and y be the number of graphs that contain p in PD and ND respectively. Then,*

$$IG(q) \le max \begin{cases} H + \frac{|PD|-x}{|D|}log\frac{|PD|-x}{|D|-x} + \frac{|ND|}{|D|}log\frac{|ND|}{|D|-x} \\ H + \frac{|ND|-y}{|D|}log\frac{|ND|-y}{|D|-y} + \frac{|PD|}{|D|}log\frac{|PD|}{|D|-y} \\ IG(p). \end{cases} \quad (3)$$

where $H = -(\frac{|PD|}{|D|}log\frac{|PD|}{|D|} + \frac{|ND|}{|D|}log\frac{|ND|}{|D|})$.

Theorems and inequalities presented above can be directly applied in the first iteration of greedy selection. For later iterations of greedy selection, we can define a similar upper bound for pruning. For this goal, we first give the concept of equivalence class.

Definition 8 (Equivalence Class). *Let D be a graph database, G be a graph in D, and T be a set of graph patterns. Then, the equivalence class of G w.r.t T is defined as* $\{G^{'}|G^{'} \in D, \forall p \in T, I(p \subseteq G) = I(p \subseteq G^{'})\}$, *where* $I(.)$ *is the indicator function.*

Based on the above definition, we know that a set of graph patterns *T* can partition a graph database *D* into a set of equivalence classes (or blocks) $D_T = \{B_i | 1 \le i \le l\}$ such that $D = \bigcup_{1 \le i \le l} B_i$.

Using the above concept of Equivalence Class, we can obtain the pruning condition for the entropy measure when the set of selected graph patterns is not empty, as shown in Theorem 6. The pruning condition for the information gain measure is shown in Theorem 7.

Theorem 6. *Let D be a graph database, T be the set of selected patterns,* D_T *be the set of equivalence classes w.r.t. T and D,* $p \subseteq q(p \notin T, q \notin T)$*. Then,*

$$H(T \cup \{q\}) \le \textstyle\sum_{B_i \in D_T} \begin{cases} -\frac{|B_i|}{|D|}log\frac{|B_i|}{2|D|}, & \text{if } supp_{B_i}(p) \ge 1/2 \\ H_{B_i}(p), & \text{if } supp_{B_i}(p) < 1/2 \end{cases} \quad (4)$$

where $H_{B_i}(p) = -(\frac{x_i}{|D|}log\frac{x_i}{|D|} + \frac{|B_i|-x_i}{|D|}log\frac{|B_i|-x_i}{|D|})$*, where* x_i *is the number of graphs that contain p in* B_i.

Theorem 7. *Let D=PD+ND, where PD and ND are positive and negative graph databases respectively, T be the set of selected patterns,* D_T*be the set of equivalence classes w.r.t. T and D,* $p \subseteq q(p \notin T, q \notin T)$*. Then,*

$$IG(T \cup \{q\}) \le H + \textstyle\sum_{B_i \in D_T} max \begin{cases} \alpha_{B_i} \\ \beta_{B_i} \\ \gamma_{B_i} \end{cases} \quad (5)$$

where $H = -(\frac{|PD|}{|D|}log\frac{|PD|}{|D|} + \frac{|ND|}{|D|}log\frac{|ND|}{|D|})$, $\alpha_{B_i} = \frac{m_i-x_i}{|D|} log\frac{m_i-x_i}{|B_i|-x_i} + \frac{n_i}{|D|}log\frac{n_i}{|B_i|-x_i}$, $\beta_{B_i} = \frac{n_i-y_i}{|D|}log\frac{n_i-y_i}{|B_i|-y_i} + \frac{m_i}{|D|}log\frac{m_i}{|B_i|-y_i}$, $\gamma_{B_i} = (\frac{x_i}{|D|}log\frac{x_i}{x_i+y_i} + \frac{y_i}{|D|}log\frac{y_i}{x_i+y_i}) + (\frac{m_i-x_i}{|D|}log\frac{m_i-x_i}{|B_i|-(x_i+y_i)} + \frac{n_i-y_i}{|D|}log\frac{n_i-y_i}{|B_i|-(x_i+y_i)})$, *where* m_i *and* n_i *are the number of positive and negative graphs in* B_i *respectively, and* x_i *and* y_i *are the number of positive and negative graphs that contain p in* B_i *respectively.*

Our pruning techniques introduced above can be integrated into the framework of any frequent subgraph mining algorithm. In this subsection, we integrate the pruning techniques into the DFS code tree enumeration framework adopted by gSpan [11], and develop an efficient algorithm, DM-TopK, to mine a set of top-k graph patterns jointly maximizing some significance measure from graph databases. The pseudo-code for the DM-TopK algorithm is shown in Algorithm 2.

We now explain the major steps of the DM-TopK algorithm. In the preprocessing stage, DM-TopK first removes the infrequent vertexes and edges from the input graph database, and then obtains the set of frequent edges. Then, DM-TopK starts an iteration procedure. At each iteration, DM-TopK performs the depth-first search in the graph pattern space for each frequent 1-edge graph pattern, and outputs a graph pattern *p* such that $M(T\cup\{p\})$ is maximized.

In Line 1 of Function *MiningNextPattern*, *p*≠*min*(*p*) prunes duplicate subgraphs and all their descendants, where *min*(*p*) is the minimum DFS code of *p*. Please refer to [11] for more details. We use global variable *best* to denote the best graph pattern selected so far. Line 3-8 deals with the case that the current set of selected graph patterns *T* is empty. If the current pattern *p* is better than *best* based on the greedy rule in Line 4, we replace *best* with *p* in Line 5. We compute the upper bound *ub* of *M*(*q*) for any supergraph *q* of *p* using Theorem 4, 5 or Corollary 1 in Line 6. If the *M*(*best*) is larger than *ub* in Line 7, then the branch rooted at *p* cannot contain any graph pattern *q* such that *M*(*q*) is larger that *M*(*best*), and thus we can safely prune the branch. Similarly, Line 9-14 deals with the case that the current set of selected graph patterns *T* is not empty based on Theorem 6 and 7. In Line 15, we scan the graph database *D* once, and enumerate all frequent supergraphs $p \diamond_r e$ of *p* where *p* can be right-most extended to $p \diamond_r e$. Line 16-17 recursively invoke Function *MiningNextPattern* to continue the enumeration process for each right-most extended supergraph of *p*. gSpan [11] has shown that performing the right-most extension only in the minimum DFS code can guarantee the completeness of mining results. More details can be found in [11].

Once *T* contains *k* graph patterns, DM-TopK terminates. Since the solution generated by DM-TopK is same as that generated by PP-TopK, the offline approximation bound for the entropy measure and the online approximation bound for the information gain still hold for DM-TopK.

Some optimization strategies can be applied in DM-TopK. For example, at each iteration, we also maintain the search space built by a DFS Code Tree in previous iteration as long as memory allows. This substantially speeds up the whole mining procedure, since DM-TopK will reduce the cost of scanning the graph database at each iteration. If the size of the DFS tree is too large to be stored fully in memory, we can only store several layers of the whole DFS tree to facilitate efficient mining.

Algorithm 2: **DM-TopK**

Input: (1) A graph Dataset *D*, (2) A minimum support *min_sup*, and (3) A significance measure *M*, and (4) Number of output patterns *k*.

Output: The set of *k* graph patterns *T* that jointly maximize *M*

```
Remove infrequent vertices and edges from $D$;
$S^1$ = Minimum DFS codes of frequent edges in $D$;
$T = \emptyset$;
$best = \emptyset$;
while $|T| < k$ do
    for each code $p$ in $S^1$ do
        if $best = \emptyset$ then
            $best = p$;
        MiningNextPattern($p$;$D$;$min\_sup$;$M$;$T$;$best$);
    $T = T \cup \{best\}$;
    $best = \emptyset$;
Output $T$;
```

Function : MiningNextPattern

Input: (1) A DFS code *p*, (2) A graph Dataset *D*, (3) A minimum support *min_sup*, (4) A significance measure *M*, (5) The set of selected graph patterns *T*, and (6) The best graph pattern discovered so far *best*.

Output: The best graph pattern discovered after searching the branch rooted at *p*

```
if $p$ != $min(p)$ then
    Return;
if $|T| = 0$ then
    if $M(best) < M(p)$ then
        $best = p$;
    Compute an upper bound $ub$ for $q$ ($p \subseteq q$) using Theorem
      4 or Theorem 5;
    if $M(best) > ub$ then
        Return;
if $|T| > 0$ then
    if $M(T \cup best) < M(T \cup p)$ then
        $best = p$;
    Compute an upper bound $ub$ for $q$ ($p \subseteq q$) using Theorem
      6 or Theorem 7;
    if $M(T \cup best) > ub$ then
        Return;
Scan $D$ once, find every frequent edge $e$ such that $p$ can be
  right-most extended to$p \diamond_r e$;
for each right-most extended child $p \diamond_r e$ of $p$ do
    MiningNextPattern($p \diamond_r e$;$D$;$min\_sup$;$M$;$T$;$best$);
```

V. Experimental Results

In this section, we conduct extensive experiments to examine the quality of extracted top-k graph patterns and efficiency of the proposed methods.

A. Experimental Settings

We derived several graph datasets from a publicly available set of chemical compounds [21]. This set of chemical compounds is the AIDS antiviral screen compound dataset from developmental therapeutics program in NCI/NIH which contains about 44,000 chemical compounds. Among all AIDS chemical compounds, the results of screening tests are categorized into three classes: CA: confirmed active; CM: confirmed moderately active; and CI: confirmed inactive. 422 compounds of them belong to active, 1081 ones are in class CM, and the remaining ones are in class CI. In our experiments, we construct three graph datasets from NCI/CA, NCI/CM and NCI/CI respectively. The graph dataset associated with NCI/CI includes 5,000 compounds which are randomly sampled from more than 40,000 compounds in CI class. Each graph in NCI/CA has 40 vertexes and 42 edges on average, each one in NCI/CM has 27 vertexes and 28 edges on average, and each one in NCI/CI has 20 vertexes and 22 edges on average.

Our algorithm is implemented in C++ with STL library support and compiled by g++ with -O3 optimization. All our experiments are performed on a 3.2GHZ Intel Pentium-4 PC with 1G memory, running RedHat 8.0.

Note that PP-TopK and DM-TopK will generate the same results. Additionally, we use T-TopK to refer to the traditional method extracting top-k patterns completely based on single significance, instead of considering joint significance.

B. Quality of Top-K Patterns

In the following, we demonstrate that our top-k graph patterns that consider joint significance are superior to the top-k graph patterns that only consider single significance in terms of classification accuracy.

We formulate three classification problems on NCI dataset. In the first problem we separate active (CA) and moderately active (CM) compounds. In the second problem we separate active (CA) and inactive (CI) compounds. In the third problem we separate moderately active (CM) and inactive (CI) compounds.

We build classifiers based on different sets of top-k graph patterns. One set considers joint significance of graph patterns, which is generated by PP-TopK/DM-TopK. The other set only considers single significance of each graph patterns, which is generated by T-TopK. The information gain is used as significance measure in this experiment. We mine top-k graph patterns from positive and negative class respectively using the same minimum support 10%. Top-k graph patterns from two different class will be combined as classification features. LIBSVM [22] with default parameters is used as the classification model. The classification accuracy is evaluated with 5-fold cross validation. We use the area under the ROC curve (AUC) to measure the classification performance. A perfect model will have the area of 1. Table 1 summarizes the AUC using different sets of top-k graph patterns w.r.t. different k.

As one can see, the classification accuracy achieved by top-k patterns that consider joint significance is clearly better than that achieved by top-k patterns that only consider single significance. This is because that the set of top-k patterns that consider single significance contains many correlated features. When correlated features are used together, the overall effect (i.e. classification performance) will be reduced. Assume that a feature which can correctly predict some examples is already selected, to improve the overall accuracy, the features considered subsequently need to predict better on those examples or subspaces of the dataset that the chosen feature cannot predict correctly. The traditional top-k approach does not consider this, whereas our top-k measure does use this combined effect. This illustrates an advantage of joint significance over single significance.

To further demonstrate the effect of joint significance, we run T-TopK and DM-TopK on CA database with *min_sup* = 10% to extract top-5 graph patterns, which are shown in Fig. 1 and 2. We also run T-TopK and DM-TopK on CM database with *min_sup* = 10% to extract top-5 graph patterns, which are shown in Fig. 3 and 4.Without considering joint significance, the top-5 results returned by T-TopK overlap substantially with respect to structure, as shown in Fig. 1 and 3. Practically, if the first graph pattern is selected, all the other graph patterns become insignificant. By considering joint significance, the top-5 results returned by DM-TopK are different from each other with respect to structure, as shown Fig. 2 and 4. This again illustrates that joint significance measure is clearly superior to single significance measure for a set of graph patterns.

Table 1
Classification Performance Based on Different Top-K Graph Patterns

	CA vs. CM		CA vs. CI		CM vs. CI	
	Single Info Gain	Joint Info Gain	Single Info Gain	Joint Info Gain	Single Info Gain	Joint Info Gain
K=5	0.6032	0.7530	0.7222	0.8284	0.5069	0.5981
K=10	0.6608	0.7966	0.5699	0.9074	0.6049	0.6864
K=15	0.6847	0.8030	0.6161	0.9216	0.5872	0.7415
K=20	0.7179	0.8023	0.7635	0.9244	0.5912	0.7435
K=25	0.6952	0.7998	0.7769	0.9224	0.6201	0.7493

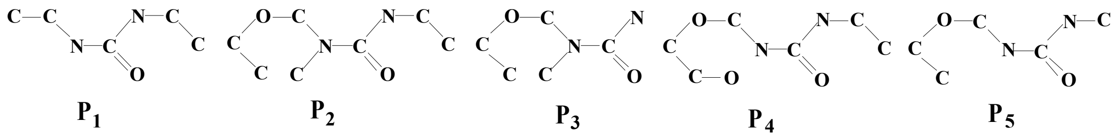


Fig. 1. Top-5 graph patterns generated by the traditional method T-TopK on CA dataset(*min_sup*=10%)

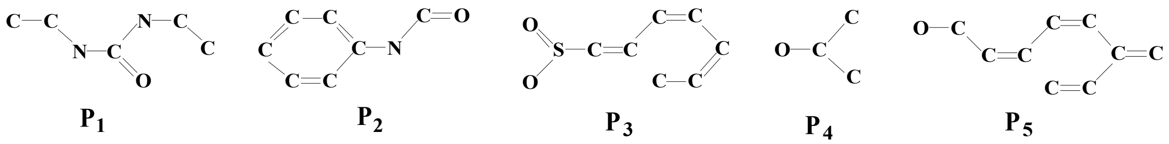


Fig. 2. Top-5 graph patterns generated by our method PP-TopK/DM-TopK on CA dataset(*min_sup*=10%)

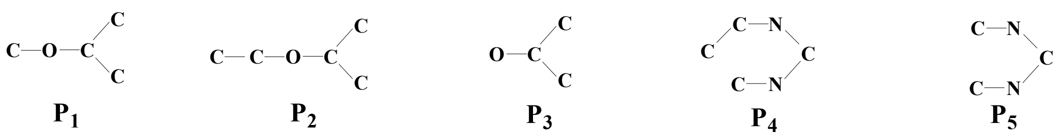


Fig. 3. Top-5 graph patterns generated by the traditional method T-TopK on CM dataset(*min_sup*=10%)

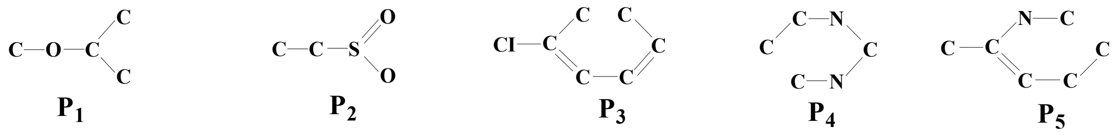
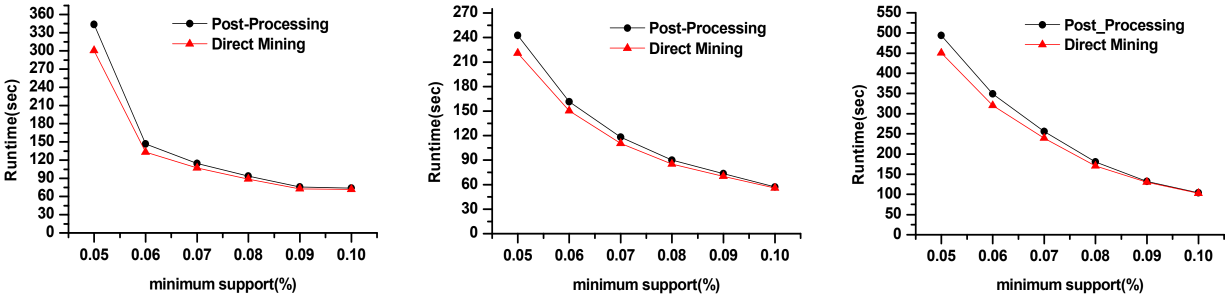


Fig. 4. Top-5 graph patterns generated by our method PP-TopK/DM-TopK on CM dataset(*min_sup*=10%)



(a): CA, K=20 (b): CM, K=20 (c): CI, K=20

Fig. 5. Post-Processing method PP-TopK vs. Direct Mining method DM-TopK

C. *Efficiency for PP-TopK and DM-TopK*

In this subsection, we study the computational performance of the post-processing method PP-TopK and direct mining method DM-TopK.

Fig. 5 compares the runtime of PP-TopK and DM-TopK with respect to different support on CA, CM and CI datasets, with a fixed k = 20. From Fig. 5, we can see that the pruning techniques in DM-TopK are very effective in improving the efficiency. It is because that when selecting a best pattern at each step, the pruning techniques can effectively prune the unpromising parts of search space and hence DM-TopK can filter a large number of graph patterns that must be explored in PP-TopK. Furthermore, we can observe the fact that the lower the minimum support threshold *min_sup*, the more effective the pruning techniques adopted in DM-TopK.

D. *Scalability*

We also study the scalability of our algorithm DM-TopK in terms of the base size of graph datasets. We replicate CA and CM datasets respectively from 1 to 10 times and run DM-TopK with fixed relative *min_sup*. The experimental results are shown in Fig. 6. It is evident that DM-TopK shows linear scalability in runtime against the number of input graphs.

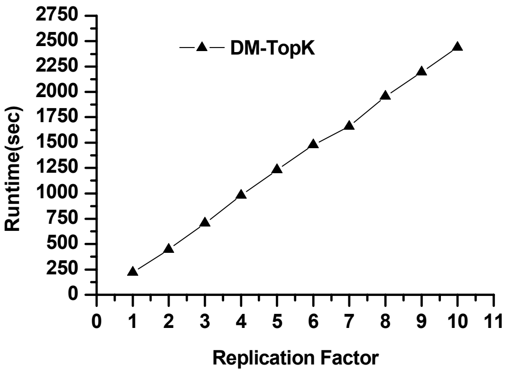


Fig. 6. Scalability Test on CM, *min sup* = 5% and *K*=20

VI. Related Work

Many frequent subgraph mining algorithms have been proposed. They can be roughly classified into two categories. Algorithms in the first category, including AGM [8] and FSG [9], use a level-wise search scheme to enumerate all frequent subgraphs based on the Apriori property. Algorithms in the second category, including Mofa [10], gSpan [11], FFSM [12] and GASTON [13], use a depth-first search to enumerate all frequent subgraphs. Due to the exponential number of frequent subgraphs, attention has been paid to closed frequent subgraphs [14] and maximal frequent subgraphs [15, 16] to reduce the number of frequent subgraphs.

In spite of many algorithms related to frequent subgraph mining, little work focuses on mining graph patterns based on other significance measure except support. Recently, He and Singh [23] introduced a statistical model to evaluate significance of subgraphs in the feature space. Yan et al. developed a framework to mine significant patterns based on structural leap search [24]. However, these techniques still consider the significance of each graph pattern independently, rather than the joint significance for a set of graph patterns.

Different from the traditional top-k method, our joint significance measure evaluates the overall significance for a set of graph patterns. As shown in our experiments, when selecting top-k graph patterns, joint significance is clearly superior to single significance.

VII. Conclusion

In this paper, we propose a novel problem of mining top-k graph patterns jointly maximizing some significance measure from graph databases. Based on entropy and information gain, we give two problem formulations, EM and IGM, and prove that they are NP-hard. We then present two efficient algorithms, PP-TopK and DM-TopK, for mining top-k graph patterns. Experimental study confirms that PP-TopK and DM-TopK are efficient and scalable in terms of the base size of input databases. The results in classifying chemical compounds demonstrate the advantage of our top-k definition against the traditional top-k definition. Since graphs represent the most general type of patterns, the concepts and methods presented in this paper can also be applied to mine other type of top-k patterns like itemsets, sequences and tree.

Acknowledgment

This work was supported in part by the National Grand Fundamental Research 973 Program of China (Grand No. 2006CB303000), the Key Program of National Natural Science Foundation of China (Grant No. 60533110), the NSF of China (Grant No. 60773063) and the Research Foundation of Harbin for Youth Innovative Talents (Grant No. 2009RFQXG080).

REFERENCES

[1] H. Berman, J. Westbrook and Z. Feng et al. “The protein data bank,” *Nucleic Acids Research*, 28:235-242, 2000.

[2] National library of medicine *http://chem.sis.nlm.nih.gov/ chemidplus*.

[3] The International Network for Social Network Analysis. *http://www.insna.org*.

[4] S. Raghavan and H. Garcia-Molina. “Representing Web graphs,” *Proc. 19th IEEE Int'l Conf. Data Engineering (ICDE '03)*, pp. 405-416, 2003.

[5] X. Yan, P. S. Yu, and J. Han. “Graph indexing: a frequent structure-based approach,” *In Proc. of the ACM SIGMOD international conference on Management of data*, pages 335-346, 2004.

[6] J. Cheng, Y. Ke, W. Ng, and A. Lu. “Fg-index: towards verification-free query processing on graph databases,” *In Proc. of the ACM SIGMOD international conference on Management of data*, pp. 857-872, 2007.

[7] M. Deshpande, M. Kuramochi, and G. Karypis. “Frequent sub-structure based approaches for classifying chemical compounds,” *Proc. 3rd IEEE Int'l Conf. Data Mining (ICDM '02)*, pp. 35-42, 2003.

[8] A. Inokuchi, T. Washio and H. Motoda. “An apriori-based algorithm for mining frequent substructures from graph data,” *Proc. 4th European Conf. on Principles and Practice of Knowledge (PKDD '00)*, pp. 13-23, 2000.

[9] M. Kuramochi and G. Karypis. “Frequent subgraph discovery,” *Proc. 1st IEEE Int'l Conf. Data Mining (ICDM '01)*, pp. 313-320, 2001.

[10] C. Borgelt and M. R. Berhold. “Mining molecular fragments: Finding relevant substructures of molecules,” *Proc. 2nd IEEE Int'l Conf. Data Mining (ICDM '02)*, pp. 51-58, 2002.

[11] X. Yan and J. Han. “gSpan: Graph-based substructure pattern mining,” *Proc. 2nd IEEE Int'l Conf. Data Mining (ICDM '02)*, pp. 721-724, 2002.

[12] J. Huan, W. Wang, and J. Prins. “Efficient mining of frequent subgraphs in the presence of isomorphism,” *Proc. 3rd IEEE Int'l Conf. Data Mining (ICDM '03)*, pp. 549-552, 2003.

[13] S. Nijssen and J. N. Kok. “A Quickstart in Frequent Structure Mining can make a Difference,” *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '04)*, pp. 647-652, 2004.

[14] X. Yan and J. Han. “CloseGraph: Mining closed frequent graph patterns,” *Proc. 9th ACM SIGKDD Int'l Conf. Knowledge Discov-ery and Data Mining (KDD '03)*, pp. 286-295, 2003.

[15] J. Huan, W. Wang, J. Prins and J. Yang. “SPIN: Mining Maximal Frequent Subgraphs from Graph Databases,” *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '04)*, pp. 581-586, 2004.

[16] L. T. Thomas, S. R. Valluri, and K. Karlapalem. “Margin: Maximal frequent subgraph mining,” *Proc. 6th IEEE Int'l Conf. Data Mining (ICDM '06)*, pp. 1097-1101, 2006.

[17] S. A. Cook. “The complexity of theorem-proving procedures,” *Proc. 3rd ACM symposium on Theory of computing(STOC '71)*, pp. 151-158, 1971.

[18] T. M. Cover and J. A. Thomas. “Elements of Information Theory,” *Wiley Interscience*, 1991.

[19] P. Tan, V. Kumar, and J. Srivastava. “Selecting the right interestingness measure for association patterns,” *In Proc. of SIGKDD*, pages 32-41, 2002.

[20] G. Nemhauser, L. Wolsey, and M. Fisher. “An analysis of the approximations for maximizing submodular set functions,” *Mathematical Programming*, 14:265-294, 1978.

[21] http://dtp.nci.nih.gov/docs/3d_database/structural_information/structural_data.html

[22] C. Chang and C. Lin. “LIBSVM: a library for support vector machines”, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[23] H. He and A. K. Singh. “GraphRank: Statistical Modeling and Mining of Significant Subgraphs in the Feature Space,” *Proc. 6th IEEE Int'l Conf. Data Mining (ICDM '06)*, pp. 885-890, 2006.

[24] X. Yan, H. Cheng, J. Han, and P. S. Yu. “Mining Significant Graph Patterns by Scalable Leap Search,” *In Proc. of the ACM SIGMOD international conference on Management of data*, pages 335-346, 2008.

Yong Liu, male, born in 1975, Ph. D. candidate. His research interests include graph mining and graph data management.

Jianzhong Li, male, born in 1950, professor, Ph.D. supervisor. His research interests include data mining, data warehouse, sensor network, grid and bioinformatics.

Jinghua Zhu, female, born in 1976, Ph.D. candidate. Her research interests include query processing and data mining in sensor network.

Hong Gao, female, born in 1966, professor, Ph.D. supervisor. Her research interests include data mining, data warehouse, sensor network.