

# GraphRank: Statistical Modeling and Mining of Significant Subgraphs in the Feature Space

Huahai He      Ambuj K. Singh  
Department of Computer Science  
University of California, Santa Barbara  
Santa Barbara, CA 93106, USA  
{huahai, ambuj}@cs.ucsb.edu

## Abstract

We propose a technique for evaluating the statistical significance of frequent subgraphs in a database. A graph is represented by a feature vector that is a histogram over a set of basis elements. The set of basis elements is chosen based on domain knowledge and consists generally of vertices, edges, or small graphs. A given subgraph is transformed to a feature vector and the significance of the subgraph is computed by considering the significance of occurrence of the corresponding vector. The probability of occurrence of the vector in a random vector is computed based on the prior probability of the basis elements. This is then used to obtain a probability distribution on the support of the vector in a database of random vectors. The statistical significance of the vector/subgraph is then defined as the  $p$ -value of its observed support. We develop efficient methods for computing  $p$ -values and lower bounds. A simplified model is further proposed to improve the efficiency. We also address the problem of feature vector mining, a generalization of item-set mining where counts are associated with items and the goal is to find significant sub-vectors. We present an algorithm that explores closed frequent sub-vectors to find significant ones. Experimental results show that the proposed techniques are effective, efficient, and useful for ranking frequent subgraphs by their statistical significance.

## 1 Introduction

Recent advances in science and technology have generated a large amount of complex data. As a powerful abstract data type, graphs are often used to represent these complex data. In the database community, graph models have been used for schema matching [1], web documents, multimedia [2], and social networks [3]. In biology, graphs have been used to represent molecular structures, protein 3D structures [4], and protein interaction networks [5].

Mining structured patterns in a collection of graphs is useful for understanding the intrinsic characteristics of scientific data. In drug development, frequent pattern mining can reveal conserved substructures in a category of medically effective chemical compounds [6]. In studies of protein interaction networks, conserved patterns in multiple species reveal cellular machinery [5]. In the analysis of protein structures, the presence of conserved subgraphs in protein contact maps can reveal evolutionarily significant patterns of chemical bonds and interactions [4].

A number of techniques have been developed to find frequent subgraphs [7, 8, 9, 10, 11, 12, 13, 14] in a transactional database, i.e., a large collection of graphs. However, the usefulness of frequent subgraph mining is limited by two factors:

1. Not all frequent subgraphs are *statistically significant*.
2. There is no way to *rank* the frequent subgraphs. This hinders the identification of subgraphs of real interest, especially when the number of discovered frequent subgraphs is large.

For illustrative purposes, consider a sample graph database shown in Fig. 1 and some frequent subgraphs shown in Fig. 2. The *support* of a subgraph is the number of graphs that contain the subgraph. A subgraph is *frequent* if its support is above a given threshold. Neither the support nor the size of a subgraph is sufficient to measure the statistical significance of a subgraph, and to rank the listed subgraphs.

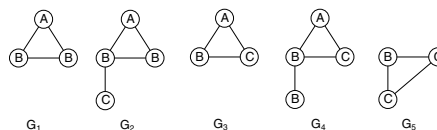


Fig. 1. A sample graph database

Subgraph	Structure	Support
$g_1$		4 ( $G_1, G_2, G_3, G_4$ )
$g_2$		3 ( $G_1, G_2, G_4$ )
$g_3$		2 ( $G_1, G_2$ )
$g_4$		2 ( $G_2, G_4$ )
$g_5$		1 ( $G_2$ )

Which subgraph is the most statistically significant?

Fig. 2. Frequent subgraphs and their supports

### 1.1 Our Approach

In this paper, we propose a technique for computing the statistical significance of frequent subgraphs. The statistical significance of a subgraph  $g$  with observed support  $\mu_0$  is defined as the probability that  $g$  occurs in a database of random graphs with support  $\mu \geq \mu_0$ , namely the  $p$ -value of  $g$ . Using this measure, we can rank the frequent subgraphs, and/or remove insignificant ones.

The main challenge of the above procedure is how to estimate the probability that a subgraph occurs in a random graph. As graphs have flexible structures, it is difficult to estimate such probability directly in the graph space (Note that the problem of determining whether a graph is a subgraph of another is NP-complete). Milo et al [15] adopted a simulation approach: generate many random graphs while maintaining some empirical measures such as degree of vertices, number of edges, and then count the ones that contain the subgraph. However, this approach is neither scalable to a large collection of graphs nor precise for computing and comparing small  $p$ -values.

We address the above challenge by transforming graphs into a feature space (Fig. 3). First, we use domain knowledge to define a set of basis elements such as vertices, edges, or small subgraphs. A graph is simply regarded as a collection or a histogram of basis elements; this defines its feature vector. Then, we approximate the question of significance of a subgraph by considering the significance of its feature vector in the feature space (Fig. 4). This is a simpler problem that admits closed-form solutions. Although structural information of a graph is lost in the feature space, statistics on the basis elements are still captured. As shown by the experimental results, this approximation is suitable for the discovery of significant subgraphs.

In the second half of the paper, we address the problem of feature vector mining, a simplified version of graph mining. Vector (aka histogram and multiset) mining is an im-

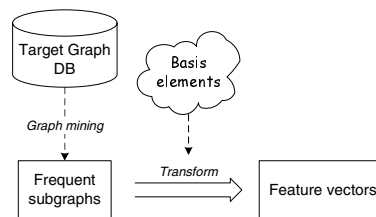


Fig. 3. Represent graphs as feature vectors

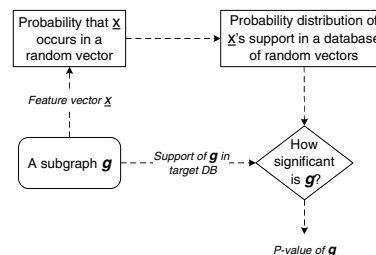


Fig. 4. Compute  $p$ -value of a subgraph

portant generalization of frequent itemset mining. We develop **ClosedVect**, an algorithm that explores *closed* subvectors to find significant ones. We prove that **ClosedVect** is optimal in terms of the number of search states.

We validate the quality of our technique through experiments on chemical compounds and synthetic graphs. In particular, we find that a specific subgraph, neither largest nor most frequent, turns out to be the largest common subgraph in a specific class of medically effective compounds. This finding validates the practical usefulness of our approach. We also demonstrate the efficiency of the computational methods and the feature vector mining algorithm.

The main contributions of our work are as follows:

1. We propose a technique for computing the  $p$ -values of frequent subgraphs, and show that frequent subgraph can be ranked by this measure.
2. We address the problem of feature vector mining, and present an algorithm for mining significant closed subvectors. This is an important problem in its own right.

The remainder of the paper is organized as follows. Section 2 discusses how to represent graphs as feature vectors. Sections 3 and 4 present a probabilistic model and a simplified model. Section 5 describes the feature vector mining. Experimental results are reported in Section 6. We conclude with a brief discussion in Section 7.

## 2 Representing Graphs as Feature Vectors

We view a graph as a collection of basis elements  $\mathbb{B} = \{\hat{b}_1, \dots, \hat{b}_m\}$ . These basis elements can be vertices, edges, or small graphs. Each basis element  $\hat{b}_i$  is associated with a *prior probability*  $\theta_i$ . We first discuss how to select basis elements and transform graphs into feature vectors.

## 2.1 Feature Selection

The selection of basis elements is application-dependent and may require domain knowledge. A basic approach is to select all types of vertices or edges as features. The drawback of this approach is that it does not capture any structural information of graphs.

For other graphs such as chemical compounds, one may choose small graphs such as Benzene rings. In this case, the number of small graphs could be large and they may overlap structurally. Thus, selecting a representative subset would be more appropriate. The following criteria for selection can be used: 1) frequency: frequent basis elements are more representative of graphs; 2) size: large basis elements carry more structural information (but would be less frequent); 3) structural overlap: overlapping basis elements are relatively not independent; 4) Co-occurrence: basis elements that frequently occur together are not independent. Based on these criteria, one may select basis elements by a greedy approach [16]: choose the  $k^{th}$  best element according to its benefit gained (e.g., frequency) and its relevance (e.g., overlap, covariance) to the previously selected  $k - 1$  basis elements.

For the sample database shown in Fig. 1, we use all kinds of edges as the basis, i.e.,  $\mathbb{B} = \{A-B, A-C, B-B, B-C, C-C\}$ . The prior probabilities are empirically computed using their frequency in the database, i.e.,  $\underline{\theta} = (\frac{6}{17}, \frac{2}{17}, \frac{3}{17}, \frac{5}{17}, \frac{1}{17})$ .

## 2.2 Transforming Graphs into Feature Vectors

After a basis is selected, we transform (sub)graphs into feature vectors. We denote a feature vector by  $\underline{x} = (x_1, \dots, x_m)$ , where  $x_i$  counts the frequency of feature  $\hat{b}_i$  in the graph. The size of  $\underline{x}$  is defined as  $|\underline{x}| = \sum x_i$ . Vector  $\underline{x}$  is a *sub-vector* of  $\underline{y}$  (and  $\underline{y}$  a *super-vector* of  $\underline{x}$ ) if  $x_i \leq y_i$  for  $i = 1, \dots, m$ , and is denoted by  $\underline{x} \subseteq \underline{y}$ . The *floor* of two vectors  $\underline{x}$  and  $\underline{y}$  is a vector  $\underline{v}$  where  $v_i = \min(x_i, y_i)$  for  $i = 1, \dots, m$ . The definition extends to a group of vectors. The *ceiling* of a group of vectors is defined analogously.

For example, the feature vector of subgraph  $g_3$  under the basis  $\mathbb{B}$  is  $(2, 0, 1, 0, 0)$ .

## 3 Probabilistic Model

In this section, we model the probability with which a feature vector  $\underline{x}$  (corresponding to a subgraph) occurs in a random vector (corresponding to a random graph), and the probability distribution of  $\underline{x}$ 's support in a database of random vectors. Statistical significance is obtained by comparison to its observed support.

### 3.1 Probability of occurrence of a feature vector in a random vector

We regard the basis  $\mathbb{B}$  as a set of  $m$  distinct events, one for every basis element, where basis element  $\hat{b}_i$  is associated with its prior probability  $\theta_i$ . A feature vector of a certain

size  $\ell$  is thus regarded as an outcome of  $\ell$  independent trials.

Given a feature vector  $\underline{y} = (y_1, \dots, y_m)$ ,  $|\underline{y}| = \ell$ , the probability that  $\underline{y}$  is observed in  $\ell$  trials can be modeled by a multinomial distribution:

$$Q(\underline{y}) = \frac{\ell!}{\prod y_i!} \prod_{i=1}^m \theta_i^{y_i}, \quad (1)$$

In other words, Eqn. (1) gives the probability of observing  $\underline{y}$  in a random vector of size  $\ell$ .

Let  $\underline{x}$  be the feature vector of a subgraph  $g$ . Then, the probability that  $\underline{x}$  occurs in a random vector of size  $\ell$  is a cumulative mass function (c.m.f.) of Eqn. (1):

$$P(\underline{x}; \ell) = \sum_{\underline{y} \text{ s.t. } y_i \geq x_i, |\underline{y}| = \ell} Q(\underline{y}) \quad (2)$$

In other words, this is the probability that  $\underline{x}$  occurs in a random vector of size  $\ell$ . The size constraint  $\ell$  is reasonable: the larger a random vector, the more likely that  $\underline{x}$  will occur in the vector.

For example, the feature vector of subgraph  $g_3$  is  $\underline{x} = (2, 0, 1, 0, 0)$ . The probability that  $\underline{x}$  occurs in a random vector of size 3 is  $P(\underline{x}; 3) = 0.066$ .

Eqn. (2) can be efficiently computed using a divide-and-conquer approach (see [17] for details).

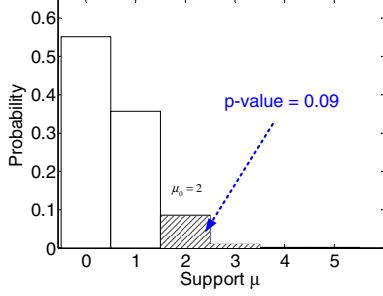
### 3.2 Probability distribution of a feature vector's support in a database of random vectors

Now we consider the support of  $\underline{x}$  in the context of a database of random vectors. This support is a random variable that follows a probability distribution. Let  $n$  be the number of vectors in the target database, we summarize the sizes of the vectors by  $\underline{\ell} = (\ell_1, \dots, \ell_d)$  and  $\underline{n} = (n_1, \dots, n_d)$ , where  $n_i$  is the number of vectors of size  $\ell_i$ , and  $\sum n_i = n$ .

If we regard a random vector as a trial, and the occurrence of  $\underline{x}$  in the vector as a "success". Then, the database of random vectors corresponds to  $n$  trials, and the support of  $\underline{x}$  corresponds to the number of successes in  $n$  trials. If the sizes of the vectors were identical, say  $\ell$ , then the support can be modeled as a binomial random variable, with parameters  $n$  and  $P(\underline{x}; \ell)$ . When the sizes are distinct, each size will correspond to one binomial random variable with parameters  $n_i$  and  $P(\underline{x}; \ell_i)$ . Then, the support of  $\underline{x}$  is the sum of the binomial random variables: the probability of  $\underline{x}$ 's support being equal to  $\mu$  is given by

$$R(\mu; \underline{x}, \underline{\ell}, \underline{n}) = \sum_{\sum t_j = \mu} \prod_{j=1}^d \text{bino}(t_j; n_j, P(\underline{x}; \ell_j)) \quad (3)$$

where  $\text{bino}(t; n, p) = \binom{n}{t} p^t (1-p)^{n-t}$  is the binomial probability distribution. In other words, the  $j^{th}$  binomial contributes  $t_j$  successes, with the sum of them equal to  $\mu$ . All



**Fig. 5. Probability distribution of  $g_3$ 's support and its p-value**

possible combinations of  $t_j$  give the total probability of observing  $\mu$ .

For the sample database of Fig. 1, a random database would have  $\underline{\ell} = (3, 4)$  and  $\underline{n} = (3, 2)$ . Fig. 5 plots the probability distribution of  $g_3$ 's support in the random database.

Eqn. (3) can be efficiently computed using a divide-and-conquer approach (see [17] for details).

### 3.3 Statistical Significance of a Feature Vector

Let  $\mu_0$  be the observed support in the target database. Then, the p-value, i.e., the probability of observing a support of at least  $\mu_0$  in the random database, is given by

$$R(\mu \geq \mu_0; \underline{x}, \underline{\ell}, \underline{n}) = \sum_{\mu=\mu_0}^n R(\mu; \underline{x}, \underline{\ell}, \underline{n}). \quad (4)$$

The smaller the p-value, the more statistically significant is the feature vector.

	$\bar{\mu}$	$\mu_0$	p-value
$g_1$	3.84	4	0.67
$g_2$	1.65	3	0.20
$g_3$	<b>0.55</b>	<b>2</b>	<b>0.09</b>
$g_4$	0.85	2	0.20
$g_5$	0.16	1	0.15

**Table 1. P-values of the subgraphs in Fig. 2; subgraph  $g_3$  has the smallest p-value.**

Now, we are ready to answer the question regarding significance raised in Fig. 2. Table 1 shows the p-values of the subgraphs as well as their expected supports. Among the subgraphs,  $g_3$  has the smallest p-value. Thus, we can claim that  $g_3$  is the most statistically significant (though it is neither the largest nor the most frequent).

## 4 A Simplified Model

In this section, we present a simplified model in which the computation of p-values is much more efficient. First, we

relax the constraint on the size of random vectors, and consider the probability that a sub-vector occurs in a random vector of arbitrary size. The probability can be written as

$$P(\underline{x}) = P(Y_1 \geq x_1, \dots, Y_m \geq x_m) \quad (5)$$

Further, if we assume that different types of basis elements are orthogonal, then the above joint probability can be decoupled into a product of probabilities:

$$\hat{P}(\underline{x}) = \prod_{i=1}^m P(Y_i \geq x_i) \quad (6)$$

where  $P(Y_i \geq x_i)$  is the probability that element  $\hat{b}_i$  occurs at least  $x_i$  times in a random vector. Since  $\hat{P}(\underline{x})$  is fixed, we then model the support of  $\underline{x}$  by a single binomial distribution, with parameters  $n$  and  $\hat{P}(\underline{x})$ .

Under this model, we compute the p-value as follows.

1. Empirically obtain the prior probabilities  $P(Y_i \geq j)$  for every basis element  $\hat{b}_i$  and every  $j$  (up to the maximum possible value). For example, element  $\hat{b}_1 = \text{"A-B"}$  occurs twice ( $G_1$  and  $G_2$ ) in the sample database, thus  $P(Y_1 \geq 2) = \frac{2}{5}$ .
2. Compute  $\hat{P}(\underline{x})$  using Eqn. (6). For subgraph  $g_3$ ,  $\underline{x} = (2, 0, 1, 0, 0)$ . Thus  $\hat{P}(\underline{x}) = P(Y_1 \geq 2) \times P(Y_3 \geq 1) = \frac{2}{5} \times \frac{3}{5} = \frac{6}{25}$ .
3. Compute the p-value of  $\underline{x}$  by  $\sum_{\mu=\mu_0}^n \text{bino}(\mu; n, \hat{P}(\underline{x}))$ , or equivalently by the regularized Beta function  $I(\hat{P}(\underline{x}); \mu_0, n)$ . When both  $n\hat{P}(\underline{x})$  and  $n(1 - \hat{P}(\underline{x}))$  are large, the binomial distribution can be approximated by a normal distribution.

## 5 Feature Vector Mining

As frequent subgraphs are represented as feature vectors and evaluated for statistical significance, an interesting question arises: *can we directly search top-K significant sub-vectors, or sub-vectors above a significance threshold?* To our best knowledge, the problem of feature vector mining has not been addressed before. Feature vector mining is important in two aspects. First, feature vectors, also known as histograms and multisets, are common ways to summarize complex data. As a result, feature vector patterns are profiles of structured patterns, and feature vector mining can work as a foundation of structured pattern mining. Second, feature vector mining is an important generalization of the well studied frequent itemset mining: each item is now associated with a count instead of a boolean value.

We develop **ClosedVect**, an algorithm that explores frequent *closed* sub-vectors to find significant ones. The algorithm consists of two phases: exploring closed sub-vectors and evaluating the significance of a closed sub-vector.

Alg. 1 outlines the phase of exploring closed sub-vectors. The algorithm explores closed sub-vectors in a bottom-up,

depth-first manner. At each search state, the algorithm “jumps” to a future state that has an immediately smaller supporting set along a branch (line 3). The corresponding sub-vector is then promoted as the *floor* of the supporting set (line 6). To prevent duplicates, each search state is associated with a beginning position  $b$ . Any future state must extend at a position greater than or equal to  $b$ . If an extension designated at position  $i$  results in a starting position of less than  $i$ , then it must be a duplicate extension (lines 7-8).

The evaluation phase (line 1) computes the p-value of a sub-vector and reports top-K significant ones (see [17] for details). Lines 9-10 estimate a lower bound on the p-value of the super-vectors of  $\underline{x}'$  and prune it if this bound is too high. This pruning is discussed further in [17].

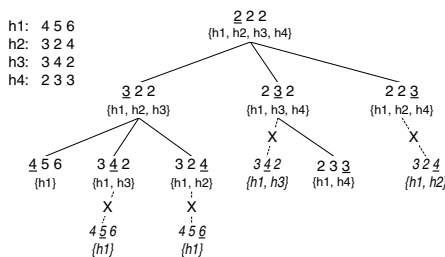
**Alg. 1** ClosedVect( $\underline{x}, \mathbb{S}, b$ )

```

 $\underline{x}$ : current sub-vector;
 $\mathbb{S}$ : supporting set of  $\underline{x}$ , i.e., vectors that contain  $\underline{x}$ ;
 $b$ : beginning position at which bins can be extended;
1: Eval( $\underline{x}, |\mathbb{S}|$ );
2: for  $i := b$  to  $m$  do
3:    $\mathbb{S}' \leftarrow \{\underline{Y} \mid \underline{Y} \in \mathbb{S}, Y_i > x_i\}$ ;
4:   if  $|\mathbb{S}'| < \text{minSupport}$  then
5:     continue;
6:    $\underline{x}' := \text{floor}(\mathbb{S}')$ ;
7:   if  $\exists j < i$  such that  $x'_j > x_j$  then
8:     continue;
9:   if p-value( $\text{ceiling}(\mathbb{S}'), |\mathbb{S}'|$ )  $\geq \text{maxPvalue}$  then
10:    continue;
11:   ClosedVect( $\underline{x}', \mathbb{S}', i$ );

```

Fig. 6 shows a running example of Alg 1. The underlined number denotes the beginning position  $b$ . Duplicate search states are pruned. For instance, the extension to “2 3 2” at position “3” leads to a supporting set “{ $h_1, h_3$ }”, of which the *floor* is “3 4 2”. This extension violates the search order and is pruned (lines 7-8).



**Fig. 6. A running example of ClosedVect**

An algorithm is *complete* if it finds all answers. It is *compact* if every search state finds at least one distinct answer. It is *duplicate-free* if it has no duplicate search states or duplicate answers. The following theorem shows the correctness and efficiency of ClosedVect (see [17] for proof).

**Theorem 1.** (Correctness and Efficiency of ClosedVect) Algorithm ClosedVect explores closed and only closed sub-vectors. It is complete, compact, and duplicate-free.

In other words, ClosedVect is optimal in terms of the number of search states because every search state corresponds to a distinct closed sub-vector.

**6 Experimental Results**

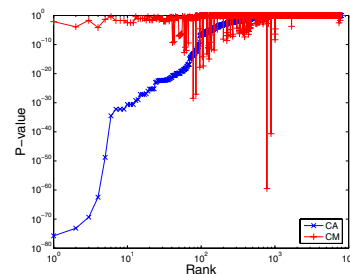
In this section, we report experimental results that validate the quality and efficiency of the proposed techniques. We use the NCI AIDS Antiviral Screen chemical dataset and focus on the category of confirmed active (CA) which contains 422 compounds. The p-value computation and the feature vector mining algorithm were implemented in Java using Sun JDK 1.5.0. All experiments were performed on an Intel 2.8GHz, 1G memory running MS Windows XP Pro.

We also experimented with a synthetic dataset and a web page visits dataset. The complete results appear in the full paper [17].

**6.1 Evaluating the Quality of the Results**

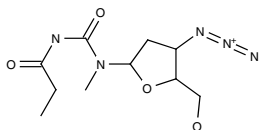
We generate two sets of basis elements to transform subgraphs into feature vectors. The first set consists of all different edges (39 in total), namely *1-edge basis*. The second set is constructed by selecting 30 out of all possible subgraphs of 3 edges (322 in total) using the greedy approach in Section 2. We call this the *3-edge basis*. For each case, we compute the prior probabilities using their frequency in the background dataset (42,000 compounds in total). Then, we use CloseGraph [10] to find frequent closed subgraphs from the CA category. With  $\text{minSup} = 5\%$ , 7879 closed subgraphs are discovered. For each subgraph, we compute its p-value using the two bases and the two models (exact and simplified) respectively.

Fig. 7 shows the p-values of the subgraphs vs. their ranks using 3-edge basis and the exact model. We also compute their p-values in the category of confirmed moderately active (CM) for cross-validation. As shown in the figure, the p-values of the subgraphs are much smaller than they would be in the context of the CM category. Further, a large number of the subgraphs are statistically insignificant. Using a



**Fig. 7. P-value vs. rank with cross-validation**

p-value cutoff, say 0.01, we are able to reduce the number of discovered subgraphs by one order of magnitude.



**Fig. 8. The most significant subgraph in CA**

Fig. 8 shows the most significant subgraph found in our results. We found that this subgraph is the largest common subgraph in the chemical class of Azido Pyrimidines<sup>1</sup>. The AZT compound, a super graph of this subgraph, is ranked 3<sup>rd</sup> in the exact model and 2<sup>nd</sup> in the simplified model. The compound has been widely used for HIV inhibition. The findings validate the practical usefulness of our approach.

We compare our ranking with a naive ranking approach: rank by size (in case of tie, rank by support). Table 2 shows the rankings of three subgraphs: the most significant subgraph (AZT\*), the largest subgraph, and Benzene. There is no current scientific evidence regarding the importance of the largest subgraph. As shown in the table, ranking by p-value is much more appropriate than the ranking by size.

Subgraph	Support	Size	Rank by p-value				Rank by size
			3-edge basis		1-edge basis		
			strict	simpl.	strict	simpl.	
AZT*	15%	19	1 <sup>st</sup>	1 <sup>st</sup>	40 <sup>th</sup>	69 <sup>th</sup>	428 <sup>th</sup>
Largest	5%	34	914 <sup>th</sup>	142 <sup>nd</sup>	752 <sup>nd</sup>	751 <sup>st</sup>	1 <sup>st</sup>
Benzene	70%	6	886 <sup>th</sup>	1424 <sup>th</sup>	6820 <sup>th</sup>	1875 <sup>th</sup>	6969 <sup>th</sup>

**Table 2. Ranking by different approaches**

## 6.2 Feature Vector Mining

We evaluate the performance of algorithm *ClosedVect* using the feature vectors of the chemical compounds. The experimental settings are:  $\text{minSupport} = 5\sim 25\%$ ;  $K = +\infty$ ;  $\text{maxPvalue} = 1, 0.01$ .

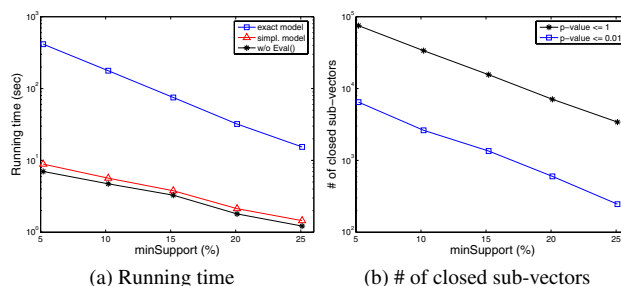
Fig. 9(a) shows the running time of *ClosedVect* w.r.t.  $\text{minSupport}$ . The running time without p-value evaluation is only in seconds. This demonstrates the high efficiency of *ClosedVect* in exploring closed sub-vectors. With p-value computation, the simplified model adds a little amount of overhead, which is much less than that of the exact model.

Fig. 9(b) shows the number of closed sub-vectors w.r.t.  $\text{minSupport}$  under the exact model. With the maximum p-value threshold set at 0.01, the number of closed sub-vectors is reduced by one order of magnitude.

## 7 Conclusions

Statistical significance and ranking are useful in the post-processing of data mining results. In this paper, we proposed a technique for evaluating the significance of frequent

<sup>1</sup><http://dtp.nci.nih.gov/docs/aids/searches/list.html>



**Fig. 9. ClosedVect on chemical compounds**

subgraphs in the feature space. We also addressed the problem of feature vector mining, and developed an algorithm that efficiently searches significant closed sub-vectors. Experimental results validated the quality, performance, and practical usefulness of the presented techniques.

**Acknowledgements:** We would like to thank Xifeng Yan and Jiawei Han for providing the code of *CloseGraph*. The work was supported in part by NSF grants IIS-0612327 and DBI-0213903.

## References

- [1] E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.* 10(4): 334-350 (2001).
- [2] S. Berretti, A. D. Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 23, 2001.
- [3] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *KDD*, 2003.
- [4] J. Hu, X. Shen, Y. Shao, C. Bystroff, and M. J. Zaki. Mining protein contact maps. In *BIOKDD*, 2002.
- [5] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. In *Proc Natl Acad Sci*, 2005.
- [6] S. Kramer, L. D. Raedt, and C. Helma. Molecular feature mining in HIV data. In *KDD*, 2001.
- [7] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Principles of Data Mining and Knowledge Discovery*, pages 13–23, 2000.
- [8] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *ICDM*, pages 313–320, 2001.
- [9] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *ICDM*, 2002.
- [10] X. Yan and J. Han. CloseGraph: Mining closed frequent graph patterns. In *KDD*, 2003.
- [11] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *ICDM*, 2003.
- [12] J. Huan, W. Wang, J. Prins, and J. Yang. SPIN: Mining maximal frequent subgraphs from graph databases. In *KDD*, 2004.
- [13] N. Vanetik, E. Gudes, and S. E. Shimony. Computing frequent graph patterns from semistructured data. In *Proceedings of ICDM*, 2002.
- [14] N. Vanetik and E. Gudes. Mining frequent labeled and partially labeled graph patterns. In *ICDE*, 2004.
- [15] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, October 2002.
- [16] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*, chapter 5, pages 181–183. Academic press, second edition, 2003.
- [17] Huahai He and Ambuj K. Singh. GraphRank: Statistical modeling and mining of significant subgraphs in the feature space. Technical report, department of computer science, University of California at Santa Barbara, 2006.