Int. J. Mach. Learn. & Cyber. (2018) 9:1445–1455
DOI 10.1007/s13042-017-0655-y

ORIGINAL ARTICLE

Uncertain maximal frequent subgraph mining algorithm based on adjacency matrix and weight

Di Wu[1] · Jiadong Ren[2] · Long Sheng[1]

Received: 17 May 2016 / Accepted: 20 February 2017 / Published online: 18 March 2017

Abstract How to mine many interesting subgraphs in uncertain graph has become an important research field in data mining. In this paper, a novel algorithm Uncertain Maximal Frequent Subgraph Mining Algorithm Based on Adjacency Matrix and Weight (UMFGAMW) is proposed. The definition of the adjacency matrix and the standard matrix coding for uncertain graph are presented. The correspondence between the adjacency matrix and uncertain graph is established. A new vertex ordering policy for computing the standard coding of uncertain graph adjacency matrix is designed. The complexity of uncertain graph standard coding is reduced, and the matching speed of uncertain subgraph standard coding is improved. The definition of the weight of uncertain graph and the mean weight of uncertain edge is proposed. The importance of the uncertain subgraphs that meet the minimum support threshold in the graph dataset is fully considered. Finally, a depth-first search weighted uncertain maximal frequent subgraph mining algorithm is discussed. According to the limiting condition of the uncertain maximum frequent subgraph and weighed uncertain edge, the number of mining results is reduced effectively. Experimental results demonstrate that the UMFGAMW algorithm has higher efficiency and better scalability.

Keywords Adjacency matrix · Weight · Uncertain graph · Maximum frequent subgraph · Frequent subgraph mining

✉ Di Wu
bestmoogoo@163.com

[1] Hebei University of Engineering, Handan, Hebei, China

[2] Yanshan University, Qinhuangdao, Hebei, China

1 Introduction

Recently, more areas of science attempt to describe complex structure objects by graph structure [1]. In the field of biology and medicine, scientists often use the graph structure to express the internal structure of macromolecules. In the internet field, the graph structure is used to describe the link relationships and views between websites. Since graph data have been more widely applied, graph data mining has become a hot topic in data mining research [2].

In practical applications, graph data have the following characteristics [3–5]:

First, there is a large quantity amount of data. Second, the data have a fast data growth rate. The update of graph data is highly frequent. Because of these features and immature technical conditions, the accuracy of obtained graph data is poor, and uncertainty exists.

For example, the field of molecular biology utilizes graph data to describe protein and its interaction network. Protein is often denoted by vertices, and the relationship of proteins is represented by the connection between vertices. Due to the restriction of detection methods during experiments, certain parts can not be accurately detected, creating uncertainty. The graph data shows the presence of the edge based on a certain probability. In summary, uncertainty exists widely in graph data. The study of uncertain graph data has very important significance [6].

The frequent subgraph mining algorithm has been researched at home and abroad. Based on Apriori properties [7], breadth-first search algorithms AGM [8] and FSG [9] for mining frequent subgraphs were proposed. In view of the shortcoming that a large number of repeat candidate subgraphs will be produced in a breadth-first search algorithm [10], a depth-first search algorithm GSpan was presented.

To optimize the searching efficiency in a large database, FFSM [11] was considered in [12]. On the basis of GSpan and a new pruning rule, the DFS code form was modified, and the mining efficiency improved.

In [13], CloseGraph algorithm for mining closed frequent subgraph patterns was proposed. SPIN algorithm was presented in [14] to mine maximal frequent subgraph patterns. However, in this algorithm, the input data are not pruned effectively, and the data are unnecessary or repeatedly mined.

Although the above algorithms are used for mining frequent subgraphs, all of these algorithms are based on the certain graph, and they can not be directly applied to uncertain graph mining.

In an early phase, Zou, who works in the Harbin Institute of Technology, performed many studies in uncertain graph mining. There are more representative achievements related to frequent subgraph mining in uncertain graph [15–17]. In these references, based on the expected semantics and probabilistic semantics, the frequent subgraphs in uncertain graph semantics are defined formally. The computational complexity of the problem is proven and effective solution technology is proposed.

An expected support and Apriori based depth-first search mining algorithm MUSE was discussed [18]. Because an efficient expected support algorithm and the subgraph search space pruning technique were proposed, the complexity of mining uncertain frequent subgraphs has been reduced from exponential level to linear level. However, the mining efficiency in dealing with the large magnitude of the uncertain graph still needs to be improved.

In [19], an efficient k-maximal frequent pattern mining algorithm on uncertain graph databases called RAKING was presented. The computation of enumeration index levels for a possible graph is avoided, but efficiency has yet to be further improved.

An efficient way of mining frequent subgraph patterns in uncertain graph databases called MUSIC was presented by Wang [20]. The algorithm relies on the Apriori property for enumerating candidate subgraph patterns efficiently. An index is applied to reduce the cost of computing expected support.

On the basis of the discriminative subgraph and a classification of uncertain graphs, an algorithm for mining uncertain frequent subgraph named AGF was proposed by Liu [21], that can switch the problem of uncertain frequent subgraph mining to uncertain frequent items mining. The efficiency of generating uncertain frequent subgraphs is improved.

In view of the problem of calculation consumption of expected support and low time efficiency of MUSE, a method that combines classification thought with BFS thought to find frequent subgraphs called EDFS was presented by Hu [22]. To reduce the space of subgraphs, an improved GSpan algorithm was used to address uncertain graphs. Integrating classification with BFS thought, uncertain frequent subgraphs are mined. The subgraph isomorphism tests and the edge existence probability tests indicate that EDFS is more efficient than MUSE.

The existing uncertain frequent subgraph mining algorithms have two main problems. First, the mining results obtained are excessive, resulting in a serious impact on the understanding and application of the results. Since the uncertain maximum frequent subgraph entails implied all uncertain frequent subgraphs, then the problem of finding uncertain frequent subgraphs is transformed into mining uncertain maximal frequent subgraphs. All of uncertain frequent subgraphs are treated equally in traditional algorithms, however, for real data, the importance of different uncertain subgraphs often varies.

Adjacency matrix and weight based uncertain maximal frequent subgraph mining algorithm UMFGAMW is presented in this paper. The correspondence between the adjacency matrix and uncertain graph is established. To improve the matching speed of uncertain subgraph standard coding, the degree of the vertices is arranged in descending order when calculating the adjacency matrix of uncertain graph. Second, to reflect the importance of each uncertain subgraph, the definitions of the weight of the uncertain graph and the mean weight of uncertain edge are designed. Finally, to reduce the number of mining results, according to the limiting condition of the mean weight of the weighted uncertain edge and the minimum support threshold under uncertain meaning, a depth-first search weighted uncertain maximal frequent subgraph mining algorithm is proposed. Weighted uncertain maximum frequent subgraphs are mined.

The remainder of this paper is organized as follows. In Sect. 2, we describe the problem definitions. Section 3 presents the UMFGAMW algorithm. In Sect. 4, the efficiency of the proposed method is analyzed based on several experimental results. Finally, we offer our conclusions and future work in Sect. 5.

2 Problem definitions

In the uncertain data model, the world model may be the most commonly used. In this model, any valid combination of the tuples may be a world instance. The probability of an instance can be obtained by calculating the probability of the relevant tuples. In this paper, the study of the uncertain graph is based on the above model.

Assuming that the uncertain graph is represented by quintuple $UG = (V, E, \sum, L, P)$, where V is the vertex set in an undirected graph. $E \subseteq V \times V$ is a set of edges. $\sum$

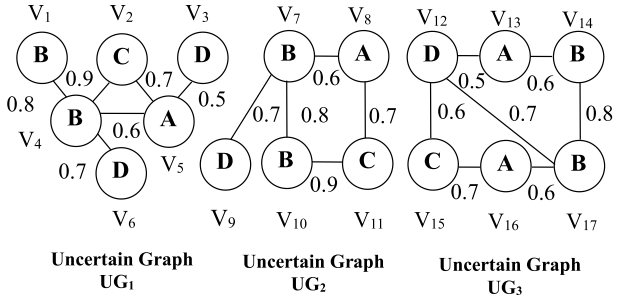

Fig. 1 The uncertain graph dataset *UGD*

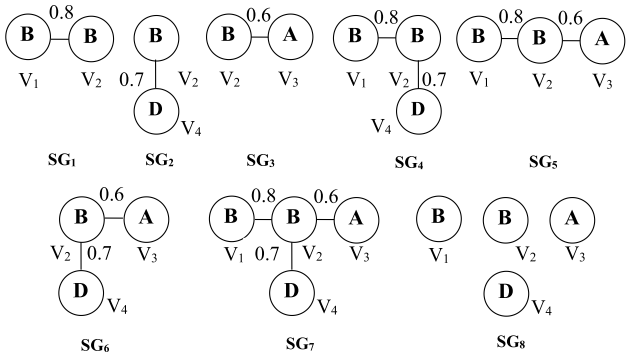

Fig. 2 All uncertain subgraphs contain in the uncertain graph UG_1

denotes a label set. L is the distribution function of vertex label. $P{:}E \rightarrow (0, 1]$ is the probability function of edge existence. For example, the existence probability of any edge e_i is denoted by $P(e_i)$, and $0 < P(e_i) \leq 1$.

Let the uncertain graph dataset $UGD = \{UG_1, UG_2, \ldots, UG_m\}$ consist of lots of uncertain graphs, where m represents the number of the uncertain graphs in UGD. The graph structure denotes the topology information. The existence probability of each edge is represented by a probability value between 0 and 1. For example, in Fig. 1, the weight of edge (B-B) in UG_1 is 0.8. It indicates that the edge e_1 exists with the probability 0.8. For an uncertain graph UG, if it has the existence probability $P(e_1) = 1$, then it indicates that the edge e_1 must exist. The certain graph is a special form of the uncertain graph. Therefore, it can be denoted by the four-tuple $G = (V, E, \sum, L)$.

The two uncertain graphs $UG = (V, E, \sum, L, P)$ and $UG' = (V', E', \sum', L', P')$ are subgraph isomorphism, if and only if UG and UG' meet a single shot function $f{:}V \rightarrow V'$, referred to as $UG \subseteq UG'$. Thus, the following two conditions are met: (1) $\forall v \in V, l(v) = l'(f(v))$ (2) $\forall (u, v) \in E, (f(u), f(v)) \in E'$.

If UG and UG' meet $UG \subseteq UG'$ and $|V_{UG}| \neq |V_{UG'}|$, then we say UG is really a subgraph isomorphism to UG', it is recorded as $UG \subset UG'$.

Assuming that graph USG is the uncertain subgraph of UGD, if and only if USG is isomorphic to any uncertain subgraph USG which is contained in at least one uncertain graph UG in UGD.

Suppose that the uncertain graph UG_r has $|L(UG_r)|$ edges, then the number of uncertain subgraphs contained in UG_r is $2^{|L(UG_r)|}$, where r is the serial number of the uncertain subgraph [21]. Here, as a result UG_1 has three edges, so the number of uncertain subgraphs in UG_1 is $2^{|L(UG_r)|} = 2^3 = 8$. All uncertain subgraphs contained in the uncertain graph UG_1 is shown as Fig. 2.

In this paper, marking all the vertices and edges that are in uncertain graph, distinguish different vertices and edges. The vertices or edges labelled with the same mark are allowed. The subgraph mining in this paper is for a marked and undirected connected uncertain subgraph.

Given that graph dataset $GD = \{G_1, G_2, \ldots, G_n\}$, the definition of the support of the subgraph SG is shown as follows:

$$Sup(SG) = \frac{\text{The number of SG in GD}}{\text{The number of graphs in GD}} \quad (1)$$

If $Sup(SG) \geq Minsup$, then SG is a frequent subgraph, where $Minsup$ is a given minimum support threshold. The value of $Minsup$ is based on experience, it is obtained through several experiments to test the best value.

3 UMFGAMW algorithm

3.1 Pre-processing uncertain data

In the UMFGAMW algorithm, there are three steps. First, the uncertain graph data is abstracted to certain space. A pretreatment process is achieved by using a frequent subgraph mining algorithm in existing certain graph data. The search space is reduced effectively. Next, the uncertainty of frequent subgraph is recovered. The expected supports of uncertain frequent subgraph patterns are calculated to narrow the search space further. The pre-treated uncertain graph results are obtained. The definition of uncertain standard matrix is introduced to establish the correspondence between the adjacency matrix and uncertain graph. The uncertain frequent subgraph is represented normally. Finally, using a depth-first search weighted uncertain maximal frequent subgraph mining algorithm, weighted uncertain maximum frequent subgraphs are mined.

3.1.1 Data pre-processing under certain meaning

In the process of pre-processing, the uncertain graph dataset is converted to certain graphs. The certain graph mining algorithm is used to delete the subgraph with support less

than *Minsup*. The search cost of the next step is reduced. In this section, the GSpan subgraph mining algorithm that is widely recognized in for efficiency and stability is utilized to obtain certain graphs. The support of each frequent subgraph in certain meaning is recorded during the process of graph pre-processing.

Different from previous work, in the traditional uncertain graph mining method, the uncertain graphs are enumerated in the possible world, and further mined in each possible world. In this section, the uncertain meaning of graph is ignored. The subgraph that is not frequent is pruned effectively. Next, the uncertain meaning of each frequent subgraph is further recovered.

3.1.2 Recovering frequent subgraph uncertainty

The uncertainty of the obtained frequent subgraph is recovered in this section. The expected support of each uncertain subgraph is calculated. For uncertain subgraph *USG*, the expected support *Sup*(*USG*) is obtained by computing the product of each subgraph existence probability and support *Sup*(*SG*) in certain meaning.

$$Sup(USG) = \prod_{l_i \in USG} USGP_{l_i} * Sup(SG) \tag{2}$$

Where, the probability calculation method of each uncertain subgraph $USGP_{l_i}$ can refer to [20].

The description of algorithm Preprocessing Uncertain Dataset Algorithm (PUDA) is shown as below.

Algorithm 1. PUDA

Input: The uncertain graph dataset *UGD*, Minimum support threshold *Minsup*, Minimum support threshold under uncertain meaning *UMinsup*.

Output: The pretreated uncertain graph dataset *PUD*.

Step1: The existence probability of each edge in the uncertain graph dataset *UGD* is set to 1, the uncertain graph dataset is translated to certain graphs;

Step2: GSpan algorithm is applied to process certain subgraphs. The support $Sup(SG_r)$ in certain meaning is stored in a subgraph. If $Sup(SG_r) < Minsup$, then SG_r is not a frequent subgraph, it is deleted directly;

Step3: The uncertainty of a frequent subgraph is restored. The support degree of each uncertain frequent subgraph is calculated by formula (2). If the corresponding value is less than *UMinsup* in the uncertain meaning, then it is pruned;

Step4: The pre-processed uncertain dataset *PUD* is obtained.

In algorithm PUDA, through certain and uncertain environment, the subgraph that is not frequent is pruned, and the search space is greatly reduced. *UMinsup* is obtained through several experiments to test the best value. Through analysing the computational complexity of the PUDA, the time complexity of the GSpan algorithm is $O(2^n \cdot 2^n)$. The complexity of the algorithm reached the index level. However, in GSpan, based on depth-first search technology, the subgraph isomorphism list of the frequent subgraphs is preserved. The number of the subgraph isomorphism operations is reduced. The value of the parameter n in PUDA is the data scale after extracting the uncertainty. Compared with the certain subgraphs with n index level in the uncertain probability space, the scale of the index level is reduced significantly. Suppose that n' is the number of the obtained uncertain frequent subgraphs. For PUDA, in addition to the time cost of the GSpan algorithm, the only other cost for traversing the uncertain frequent subgraph is $O(n')$.

3.2 Standard representation and calculation method for uncertain frequent subgraph

In this paper, the concept of uncertain standard matrix is introduced. The correspondence between the adjacency matrix and uncertain graph is established. In the process of mining uncertain frequent subgraphs, each uncertain graph is represented by the uncertain adjacency matrix. The purpose of mining uncertain subgraphs is not only to find frequent objects, but also to determine implicitly the relationship between these objects. In the uncertain graph, the entities are represented by vertices, and the relations among the vertices are represented by uncertain edges. Therefore, edge-center based adjacency matrix notation is used.

Definition 1 Assuming that uncertain graph $UG = (V, E, \sum, L, P)$ has n vertices, $V(UG) = \{V_1, V_2, \ldots, V_n\}$, then n order square matrix $A(UG) = (X_{ij})$ denotes the adjacency matrix of *UG*.

$$X_{ij} = \begin{cases} P(e) * e, & \text{if it has edge } e \text{ between } v_i \text{ and } v_j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Example 1 Computing the adjacency matrix of *UG* in Fig. 3.

Two adjacency matrixes of the given *UG* are shown in Fig. 4.

To improve the efficiency of the algorithm, because of the asymmetry, we can only reserve the upper triangular matrix of adjacency matrix. However, because allow the same mark appears, the uniqueness of the adjacency matrix can not be assured. As shown in Fig. 2, according to

different orders of the vertices, different adjacency matrixes can be obtained. To this end, the concept of the standard matrix coding is introduced. The correspondence between adjacency matrix and uncertain graph is established.

Definition 2 Assuming that $X = \begin{pmatrix} X_{11}, X_{12}, \ldots, X_{1n} \\ X_{21}, X_{22}, \ldots, X_{2n} \\ \vdots \quad\quad\quad \vdots \\ X_{n1}, X_{n2}, \ldots, X_{nn} \end{pmatrix}$ is the adjacency matrix of uncertain graph. The matrix coding for X is represented by $CD(X) = X_{12}X_{13}\cdots X_{1,m}X_{23}\cdots X_{2,m}\cdots X_{m-1,m}$, then the standard matrix coding of uncertain graph UG is denoted by $SC(UG) = \underset{X\in A(UG)}{Max}(CD(X))$.

According to the definitions of subgraph isomorphism and the adjacency matrix of uncertain graph, the computation of standard matrix coding of graphs is equivalent to the isomorphism of a graph. The reason is that if two graphs are isomorphic with each other, the standard matrix coding must be the same. Normally, by listing all possible coding and finding the maximum, the final standard matrix coding can be obtained. Calculation cost is higher, its complexity is $O(|V(UG|!)$, where $|V(UG)|$ is the number of vertices in the uncertain graph.

Suppose that uncertain graph $UG = (V, E, \sum, L, P)$, the degree of vertex is denoted by the number of associated edges of vertex $v(v \in V)$. The factor of the existence probability of edge should be considered. For example, in Fig. 3, V_1 and V_2, V_1 and V_4 are connected. Since the existence probability of e_2 and e_4 is 0.6, thus the degree of V_1 is 1.2.

To reduce the computational complexity of standard matrix coding and improve execution efficiency, the fast calculation strategy of standard matrix coding is proposed. When calculating the adjacency matrix of the uncertain graph, vertices are arranged from high to low order. By following Example 2, the strategy is explained.

Example 2 By using the adjacency matrix in Fig. 3 as an example, the degrees of five vertices are 1.2, 2.7, 0.7, 2 and 1.2, respectively. According to the degree of vertex from high to low order, and the strategy of lexicographic order

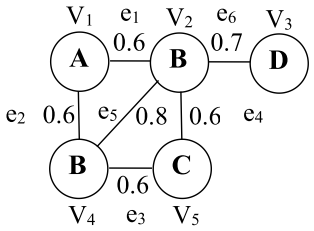

Fig. 3 The uncertain graph UG

	V_1	V_2	V_3	V_4	V_5
V_1	0	$0.6e_1$	0	$0.6e_2$	0
V_2	$0.6e_1$	0	$0.7e_6$	$0.8e_5$	$0.6e_4$
V_3	0	$0.7e_6$	0	0	0
V_4	$0.6e_2$	$0.8e_5$	0	0	$0.6e_3$
V_5	0	$0.6e_4$	0	$0.6e_3$	0

(a) The first Adjacency matrix

	V_5	V_4	V_3	V_2	V_1
V_5	0	$0.6e_3$	0	$0.6e_4$	0
V_4	$0.6e_3$	0	0	$0.8e_5$	$0.6e_2$
V_3	0	0	0	$0.7e_6$	0
V_2	$0.6e_4$	$0.8e_5$	$0.7e_6$	0	$0.6e_1$
V_1	0	$0.6e_2$	0	$0.6e_1$	0

(b) The second Adjacency matrix

Fig. 4 Two adjacency matrixes of UG

when two vertices have the same degree, the ordering result is V_2, V_4, V_1/V_5 and V_3. In this way, all possible adjacency matrixes are shown in Fig. 5.

We set the values of e_1, e_2, e_3, e_4, e_5 and e_6 are 1, 2, 3, 4, 5 and 6. According to the given values of e_1, e_2, e_3, e_4, e_5 and e_6, each matrix code can be represented by 15 columns of numbers. By sequentially comparing the numbers in each column, the matrix coding with the larger value in the front column is the largest. The matrix codings of Fig. 5a, b are $0.8e_5$ $0.6e_1$ $0.6e_4$ $0.7e_6$ 0 $0.6e_2$ $0.6e_3$ 0 0 0 0 0 0 0 0 and $0.8e_5$ $0.6e_4$ $0.6e_1$ $0.7e_6$ 0 $0.6e_3$ $0.6e_2$ 0 0 0 0 0 0 0 0, respectively. We can see that the two matrix codings are 4 0.6 2.4 4.2 0 1.2 1.8 0 0 0 0 0 0 0 0 and 4 2.4 0.6 4.2 0 1.8 1.2 0 0 0 0 0 0 0 0. According to the strategy of standard matrix coding, because the value in the second column of the matrix coding of Fig. 5b is 2.4, which is larger than the corresponding value of the matrix coding of Fig.5a. Therefore, the standard matrix coding of Fig. 3 is the same as the matrix coding of Fig.5b.

It can be seen from Example 2, using the proposed ordering strategy, the standard matrix coding is obtained by comparing codings of the two matrixes. If we enumerate all possible matrixes to calculate, 6! = 720 different matrix coding needs to be compared. It is easy to see that the uncertain graph search space is greatly reduced by ordering strategy.

According to the idea of the Apriori algorithm, when producing k-candidate uncertain subgraph, transaction is

	V_2	V_4	V_1	V_5	V_3
V_2	0	$0.8e_5$	$0.6e_1$	$0.6e_4$	$0.7e_6$
V_4	$0.8e_5$	0	$0.6e_2$	$0.6e_3$	0
V_1	$0.6e_1$	$0.6e_2$	0	0	0
V_5	$0.6e_4$	$0.6e_3$	0	0	0
V_3	$0.7e_6$	0	0	0	0

(a) The first Adjacency matrix

	V_2	V_4	V_5	V_1	V_3
V_2	0	$0.8e_5$	$0.6e_4$	$0.6e_1$	$0.7e_6$
V_4	$0.8e_5$	0	$0.6e_3$	$0.6e_2$	0
V_5	$0.6e_4$	$0.6e_3$	0	0	0
V_1	$0.6e_1$	$0.6e_2$	0	0	0
V_3	$0.7e_6$	0	0	0	0

(b) The second Adjacency matrix

Fig. 5 Two ordered adjacency matrixes of UG in Fig. 3

needed that contains the uncertain subgraph and computes the support degree. Whether the uncertain subgraph is deleted will be judged. The proposed strategy in this paper can reduce the time cost in the uncertain graph isomorphism and the calculation of support.

It is easy to see from the fast calculation strategy of standard matrix coding that the connection vertex with other vertices is reflected by the degree of the vertex. The higher the degree of the vertex, the more it connects with other vertices. Therefore, nonzero elements in the adjacency matrix should appear in front of the subgraph that is represented by the matrix coding as much as possible. In this way, the matching time for uncertain subgraph coding in transaction coding uncertain subgraph is reduced. As standard matrix coding in Example 2, most nonzero elements are concentrated in front of the coding, zero elements are at the back of the coding. The measure of efficiency for judging standard coding of the uncertain subgraph is the improvement in the substring of the uncertain graph transaction standard coding. This measure enables uncertain subgraph isomorphism to end as early as possible.

3.3 Mining weighted uncertain maximal frequent subgraphs

For traditional uncertain subgraph pattern mining algorithms, all uncertain frequent subgraph patterns will be treated equally. However, in a real dataset, different uncertain subgraph patterns often have different importance. To solve the above problem, on the basis of frequency of uncertain frequent edge, the calculation method of uncertain frequent subgraph weight is used.

Definition 3 Assume that $\{UG_1, UG_2, \ldots, UG_r\}$ is r records in uncertain graph dataset UGD. The weight of edge e_i is defined as $W(e_i) = N(e_i)/\sum_{j=1}^{r} \left|L(UG_j)\right|$, where $N(e_i) = \sum_{j=1}^{r} P(e_i) * N(e_i)_{UG_j}$ denotes the total number of appearances of edge e_i in UGD. $N(e_i)_{UG_j}$ is the number of e_i appearances in UG_j. $|L(UG_j)|$ is the number of edges in UG_j. The uncertain subgraph that contains k edges is called an uncertain k-subgraph.

For example, suppose that there are three uncertain subgraphs in UGD, they are UG_1, UG_2 and UG_3. The number of edges in UG_1, UG_2 and UG_3 are 5, 4, 6, respectively. The number of edge e_1 in UG_1, UG_2 and UG_3 are 2, 1, 2. Therefore, the weight of e_1 is $W(e_1) = (2 + 1 + 2)/(5 + 4 + 6) = 5/15 = 1/3$.

Definition 4 Assuming that the uncertain subgraph USG consists of t edges $\{e_1, e_2, \ldots, e_t\}$, then the weight of USG is defined by $W(USG) = \sum_{k=1}^{t} W(e_k)/t$.

Suppose that USG contains n different edges $\{e_1, e_2, \ldots, e_n\}$, the maximum weight and minimum weight are represented by $Max_{1\le k\le n}(W(e_k))$ and $Min_{1\le k\le n}(W(e_k))$, then the mean weight of uncertain subgraph USG is defined as follows:

$$MeanWeight(USG) = \frac{Max_{1\le k\le n}(W(e_k)) + Min_{1\le k\le n}(W(e_k))}{2} \tag{4}$$

It can be seen from Eq. (4), the mean weight of USG is described as the mean of the maximum weight and the minimum weight of all the uncertain edges in USG.

Given that uncertain subgraph USG and $MeanWeight(USG)$, the following two conditions are used to prune. (1)$Sup(USG) < Minsup$; (2)$W(USG) < MeanWeight(USG)$.

Assuming that USG is an uncertain subgraph, if it is frequent, then any $X \subseteq USG$ meets that X is an uncertain frequent subgraph. Condition (1) meets the Apriori properties. Condition (2) meets that if the weight of uncertain subgraph USG is lower than the mean weight. That is, the subgraph is not 'important', and then it will be pruned in the mining process. Direct use of these two conditions to prune, will lead to higher computational cost. More importantly, the mining results will not continually use Apriori properties to prune.

To solve this problem, the uncertain frequent 1-subgraphs are sorted according to weight from high to low order. Through mining weighted uncertain frequent subgraphs, the pruning process is accelerated and the efficiency of the algorithm is improved.

Suppose that USG is an uncertain subgraph, if $Sup(USG) * W(USG) \ge MeanWeight(USG) * Minsup$, then USG is described as a weighted uncertain frequent subgraph. If there is no uncertain weighted frequent subgraph USG', it meets that $USG \subseteq USG'$, then USG is calledthe weighted uncertain maximal frequent subgraph.

Given any two uncertain subgraphs USG and USG', the standard matrix coding $SC(USG) = P(e_1)e_1P(e_2)e_2 \cdots P(e_n)e_n$ and $SC(USG') = P(e_1)e_1P(e_2)e_2 \cdots P(e_n)e_nP(e_a)e_a$, respectively. If for any $e_i \in SC(USG)$, $W(e_a) \le W(e_i)$, and there does not exist USG with standard matrix coding $P(e_1)e_1P(e_2)e_2 \cdots P(e_n)e_nP(e_b)e_b$, then $W(e_a) \le W(e_b) \le W(e_i)$ for $\forall e_i \in SC(USG)$. USG' is denoted as a child of the USG and recorded as $USG' = CH(USG)$.

It can be seen from the definition of the child of the *USG* called *CH*(*USG*) that the proposed weighted maximal frequent subgraph mining algorithm is based on the depth-first thought. Any superset of non-weighted frequent subgraph is also a non-weighted frequent subgraph. Any subset of a weighted frequent subgraph is also a weighted frequent subgraph.

In this section, a depth-first weighted uncertain maximal frequent subgraph mining algorithm named WUMFGM is proposed.

Algorithm 2: WUMFGM(PUD, UMFGD)

Input: The preprocessed uncertain graph dataset *PUD*, Minimum support threshold *Minsup*.

Output: Uncertain Maximal frequent subgraph dataset *UMFGD*

Step1: All 1-weighted uncertain frequent subgraphs are found. Uncertain edges are sorted according to weight from high to low order. Here we set the sorted uncertain edges are $e_1, e_2, ..., e_n$;

Step2: for each e_m do

Step3: if $e_m\ e_{m+1}\ ...\ e_n$ is a weighted uncertain frequent subgraph *UMFGD* does not exist the superset of $e_i\ e_{m+1}\ ...\ e_n$

Step4: $e_m, e_{m+1}, ..., e_n$ is added to *UMFGD*;

Step5: break;

Step6: end if

Step7: if $CH(e_m) = \emptyset$ & *UMFGD* does not exist the superset of e_m then

Step8: e_m is added to *UMFGD*;

Step9: else $WUMFGM(PUD, UMFGD)$;

Step10: end for

Step11: Uncertain Maximal frequent subgraph dataset *UMFGD* is output.

In algorithm 2, all 1-weighted uncertain frequent subgraphs are found. The subgraphs are sorted in accordance with the weight from high to low order. Each 1-weighted uncertain frequent subgraph e_m is investigated one by one. If $e_m, e_{m+1}, ..., e_n$ is a weighted uncertain frequent subgraph, then we know that all subgraphs are weighted uncertain frequent subgraphs. If the superset of the uncertain subgraph in *UMFGD* does not exist, then the *UMFGD* subgraph is added to *UMFGD*. If the branch strategy is established, then the process for connecting between e_m and all subsets one by one in $e_{m+1}, ..., e_n$ is decreased. The judgement cost of weighted uncertain maximal frequent subgraphs is reduced. The search space is also greatly reduced.

Weighted uncertain maximal frequent subgraph expanding algorithm named *WUMFGE* is described as follows.

Algorithm 3: WUMFGE(PUD, UMFGD, UG)

Input: The preprocessed uncertain graph dataset *PUD*, Minimum support threshold *Minsup*.

Output: The extended weighted uncertain maximal frequent subgraph.

Step1: if $CH(UG)$ is weighted uncertain frequent subgraph then

Step2: $CH(UG)$ is added to UG;

Step3: if $CH(UG) = \emptyset$ & *UMFGD* does not exist the superset of UG then

Step4: UG is added to *UMFGD*;

Step5: end if

Step6: else

Step7: Repeat steps Step1-Step7, until depth-first extension for e_m is completed;

Step8: end if

In algorithm WUMFGE, if $CH(e_m) = \emptyset$ and the superset of e_m in *UMFGD*there does not exist, then e_m is added to *UMFGD*. Otherwise, if $CH(e_m) \neq \emptyset$, the depth-first extension for e_m is executed recursively by WUMFGE.

Uncertain maximal frequent subgraph mining algorithm based on adjacency matrix and weight called UMFGAMW is described as follows.

Algorithm 4: UMFGAMW

Input: The preprocessed uncertain graph dataset *PUD*, Minimum support threshold *Minsup*.

Output: Weighted uncertain maximal frequent subgraph dataset WUMFGD.

Step1: The uncertain graph dataset *UGD* is preprocessed by using PUDA, and then the pre-processed uncertain graph dataset *PUD* is obtained;

Step2: The correspondence between the adjacency matrix and uncertain graph is established by the uncertain standard matrix. Next, the uncertain frequent subgraph is represented normally;

Step3: By WUMFGM, all mined 1-weighted uncertain frequent subgraphs are sorted by weight from high to low order. According to the limiting condition of the mean weight of the weighted uncertain edge and the minimum support threshold under uncertain meaning, weighted uncertain maximal frequent subgraph dataset *WUMFGD* is mined.

4 Experimental results and analysis

To verify the performances of UMFGAMW, RAKING and EDFS, the experimental tests were conducted with a certain graph dataset generated from the graph data generator in [9]. Input parameters of the graph data generator are described as the following Table 1.

After generating the certain graph dataset, each edge in a certain graph is given a probability. This probability follows the normal distribution of the mean value of m and the variance of d^2.

Our experiments were run on the Intel Core 2 Duo 2.93 GHz CPU, 4GB main memory and Microsoft XP. All algorithms are written in MyEclipse 8.5. For testing the performance of UMFGAMW, we compare it with RAKING and EDFS in four aspects. The aspects are comparison of execution times of algorithms by the change of minimum support degree *Minsup*, comparison of the execution times of algorithms by the mean change of the probability of the edge in the uncertain graph, comparison of the algorithm scalability with an increasing number of uncertain graphs, and comparison of the number of the mined uncertain frequent subgraphs of the algorithm with different minimum support degree *Minsup*.

(1) Comparison of execution times of algorithms by the change of minimum support degree *Minsup*

The parameters of Data1 are $D = 10{,}000$, $L = 100$, $V = 10$, $E = 10$, $I = 5$, $T = 20$, $m = 0.9$, $d = 0.05$. The parameters of Data2 are $D = 20{,}000$, $L = 100$, $V = 10$, $E = 10$, $I = 5$, $T = 30$, $m = 0.9$, $d = 0.05$. In Data1, the execution times of the three algorithms are compared by the change of minimum support degree *Minsup*.

It is not difficult to perceive from Fig. 6, with the increase of the minimum support degree *Minsup*, that the execution time of the UMFGAMW algorithm is gradually reduced. This is because when the minimum support degree *Minsup* increases, the relative uncertain frequent subgraph is correspondingly reduced. Uncertain frequent subgraphs under a smaller threshold become less frequent and cut off. The reduction of the search space will inevitably result in a change in the search time of the algorithm, which is consistent with the theoretical results.

Figure 7 shows that the execution time is related to the number of uncertain graphs and the average size of the uncertain graphs. The larger the number, the greater the average size, and the higher the cost of the UMFGAMW algorithm. Therefore, the efficiency of the UMFGAMW in Data1 dataset is better than that in Data2.

In RAKING algorithm, K uncertain maximal frequent patterns are obtained, the parameter K is the number of the uncertain maximal frequent patterns in the original dataset. When the value of K is very large, the execution time of the RAKING will increase dramatically. If the number of the uncertain maximal frequent patterns in the original dataset is less than K, then the algorithm is automatically terminated after several iterations.

Because the uncertain subgraph pattern search space tailoring technique and database partitions are used in the EDFS algorithm, the search space is reduced to a certain extent. Therefore, in the process of searching uncertain frequent subgraphs, efficiency is higher than that in the RAKING algorithm.

For UMFGAMW, based on the frequency of uncertain frequent edges, the method of calculating the weight of the uncertain subgraph is given. First, the uncertain frequent 1-subgraphs are sorted by their weight from high to low order. By mining weighted uncertain frequent subgraphs, the process of pruning is faster. The efficiency of the algorithm is also improved. In addition, rapid calculation strategy of the standard matrix coding is proposed in the UMFGAMW. When calculating the adjacency matrix of the uncertain graph, the vertices are arranged by degree from high to low order. The matching speed of the coding of the candidate uncertain subgraph in the uncertain graph transaction coding subgraph is accelerated. Reducing the computational complexity of the uncertain standard matrix coding, improves the efficiency of the algorithm UMFGAMW.

(2) Comparison of the execution times of algorithms by the mean change of the probability of the edge in the uncertain graph

There are two possibilities for the mean change of the probability of the edge in the uncertain graph. One is change in the probability of the edge, and the other is change in the probability variance of the edge. In this paper, we compare the execution time of the algorithms in these two cases. The parameters of Data3 are $D = 10{,}000$, $L = 100$, $V = 10$, $E = 10$, $I = 5$, $T = 30$, $m1 = 0.8$, $d1 = 0.1$. The parameters of Data4 are $D = 10{,}000$, $L = 100$, $V = 10$, $E = 10$, $I = 5$, $T = 30$, $m2 = 0.9$, $d2 = 0.2$. In this experiment, $Minsup = 0.1$.

Figure 8 shows that when the possibility is gradually increased, the support degree is also increased, and the number of uncertain frequent subgraphs increases. The execution time of the algorithm is longer.

From Fig. 9, with the increase of the probability variance of the edge, the execution time of UMFGAMW is reduced. Because with the increase of the probability variance of the edge, the number of uncertain edges that have relatively low existence possibility is increased. This will reduce the number of uncertain frequent subgraphs, therefore, the running time of UMFGAMW is reduced.

Table 1 Input parameters of the graph data generator

Parameters	Meaning of the parameters
D	The number of graphs
V	The number of vertex label
E	The number of edge label
I	The average size of frequent subgraphs
L	The potential number of frequent subgraphs
T	The average size of graphs

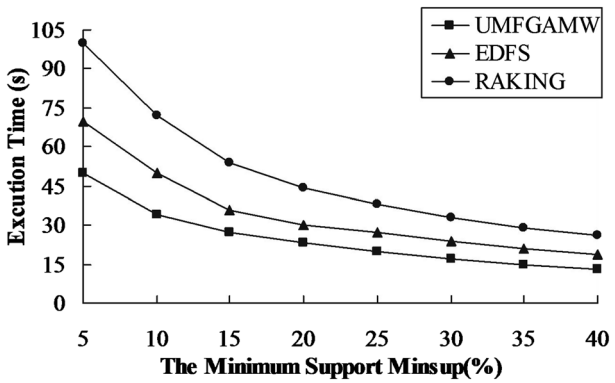

Fig. 6 Comparison of execution times of three algorithms by the change of minimum support degree *Minsup*

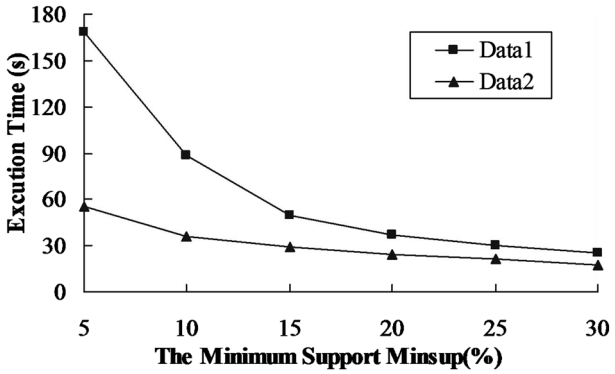

Fig. 7 Comparison of the execution time of the UMFGAMW with different datasets

(3) Comparison of the algorithm scalability with an increasing number of uncertain graphs

There are two kinds of comparison of the scalability with an increasing number of uncertain graphs. One is comparison of execution time of the UMFGAMW with a different number of uncertain graphs, and the other is comparison of the number of the mined frequent subgraph patterns of the UMFGAMW with a different number of uncertain graphs. In this paper, we compare the scalability of the UMFGAMW in these two cases. The parameters of Data5 are $D = 10{,}000$, $L = 200$, $V = 10$, $E = 10$, $I = 5$, $T = 20$, $m = 0.95$, $d = 0.05$. The parameters of Data6 are $D = 10{,}000$, $L = 100$, $V = 5$, $E = 10$, $I = 5$, $T = 20$, $m = 0.95$, $d = 0.05$. In this experiment, $Minsup = 0.1$.

Figure 10 shows that with the increase of the number of uncertain graphs, the execution time of the algorithm is longer. This is because when the number of uncertain graphs increases, the number of uncertain subgraphs needed to search is also increased.

From Fig. 11, with the increase of the number of uncertain graphs, the number of the mined uncertain frequent subgraphs is also increased. The UMFGAMW in dataset Data5 exhibits better scalability results than that in Data6.

(4) Comparison of the number of the mined uncertain frequent subgraphs of the UMFGAMW with different minimum support degrees *Minsup*

In this section, datasets Data1 and Data2 are used, for UMFGAMW algorithm, we compare the number of the mined uncertain frequent subgraphs under different minimum support degrees *Minsup*.

It is not difficult to see from Fig. 12, with the increase of support degree, the number of mined uncertain frequent subgraphs is decreased from the overall trend. The number of mined uncertain frequent subgraphs of the UMFGAMW in the dataset Data2 is less than that in Data1. This is because when the minimum support threshold *Minsup* increases, the relative uncertain frequent subgraphs are correspondingly reduced allowing frequent subgraphs to become not frequent under a small threshold. Thus, the uncertain subgraph is pruned away. The reduction of the search space will inevitably cause the corresponding reduction of search time cost. This is consistent with the theoretical results. The number of the mined uncertain frequent subgraphs is related to the number of the uncertain graphs and the average size of the uncertain graphs. If there are more uncertain graphs and they are of larger average size, there will be more mined uncertain frequent subgraphs.

5 Conclusions

In this paper, a novel algorithm UMFGAMW is proposed to resolve the problem of getting all uncertain frequent subgraphs in the traditional uncertain frequent subgraph mining algorithm. The weighted uncertain maximal frequent subgraph is obtained. Not only can it dentify important uncertain maximal frequent subgraphs, it can

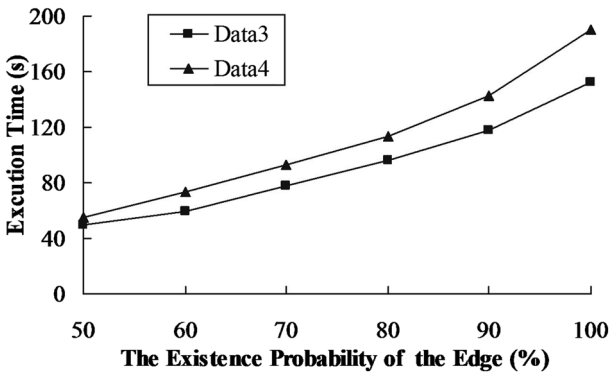

Fig. 8 Comparison of the execution time of the UMFGAMW with different probabilities of the edge

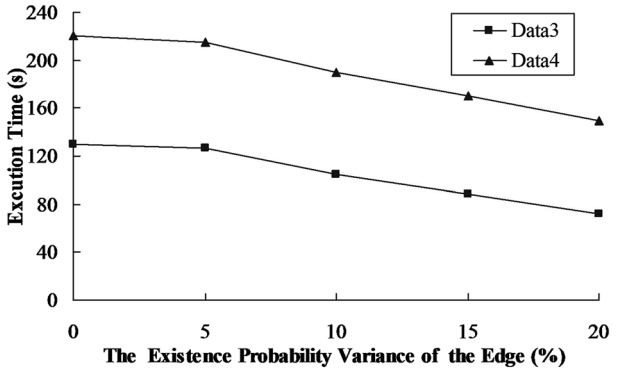

Fig. 9 Comparison of the execution time of the UMFGAMW with different probability variances of the edge

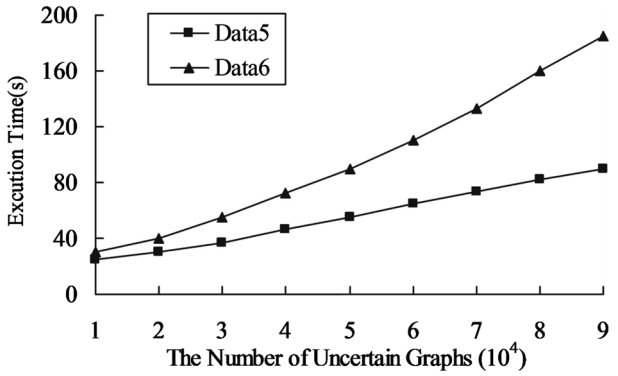

Fig. 10 Comparison of execution time of the UMFGAMW with different number of uncertain graphs

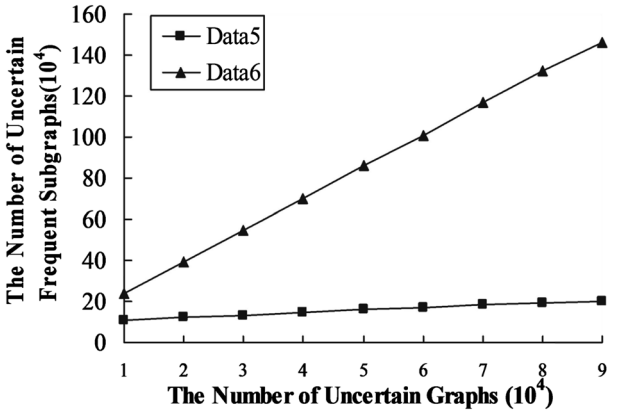

Fig. 11 Comparison of the number of the mined uncertain frequent subgraphs of the UMFGAMW with different number of uncertain graphs

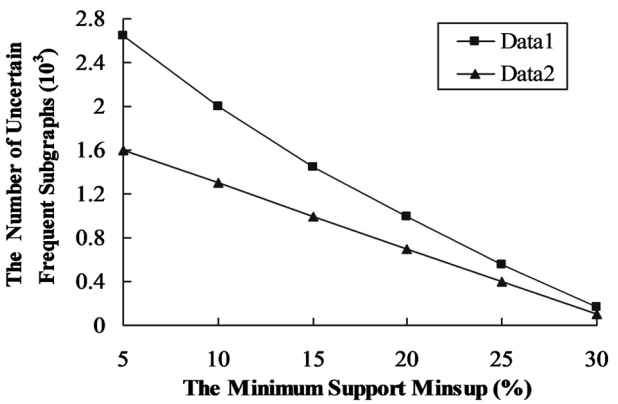

Fig. 12 Comparison of the number of the mined uncertain frequent subgraphs of the UMFGAMW with different minimum support degrees *Minsup*

also accelerate the cost of pruning. The definition of the adjacency matrix and standard matrix coding for uncertain graphs are introduced. The correspondence between the adjacency matrix and uncertain graphs is established. A new vertex ordering policy for computing the standard coding of a graph adjacency matrix is designed. Uncertain frequent subgraphs are represented normally. When calculating the adjacency matrix of the uncertain graph, vertices are arranged according to degree from high to low order. The computational efficiency of the algorithm is improved. In addition, a depth-first search weighted uncertain maximal uncertain frequent subgraph mining algorithm is proposed. All mined 1-weighted uncertain frequent subgraphs are sorted by weight from high to low order. According to the limiting condition of the mean weight of the weighed uncertain edge and the minimum support threshold under uncertain meaning, weighted uncertain maximum frequent subgraphs are obtained. The efficiency of pruning is improved. The search space of uncertain maximal frequent subgraph is reduced. The experimental results show that the proposed algorithm UMFGAMW not only can effectively reduce the number of mining results, but also has a high efficiency.

At present, the mainstream of uncertain frequent subgraph mining algorithms is primarily focused on how to improve efficiency. However, the real bottleneck of the algorithm is that the number of whole uncertain subgraphs is too high, which seriously affects the understanding and application of the mining results. Therefore, a further research direction is to explore a more efficient algorithm for simplifying the representation of uncertain frequent subgraphs and mining uncertain frequent subgraphs.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No.61170190), the Nature Science Foundation of Hebei Province (No.F2015402114, F2015402070, F2015402119) and Foundation of Hebei Educational Committee (No. YQ2014014, QN20131081). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Lee G, Yun U, Ryang H (2015) An uncertainty-based approach: frequent itemset mining from uncertain data with different item importance. Knowl Based Syst 90:239–256
2. Kurumatani N, Monji H, Ohkawa T (2014) Binding site extraction by similar subgraphs mining from protein molecular surfaces and its application to protein classification. Int J Artif Intell Tools 23:1460007
3. Douar B, Latiri C, Liquiere M et al (2014) a projection bias in frequent subgraph mining can make a difference. Int J Artif Intell Tools 23:1450005
4. Myithili K, Parvathi R, Akram M (2016) Certain types of intuitionistic fuzzy directed hypergraphs. Int J Mach Learn Cybern 7:287–295
5. NagoorGani A, Akram M, Vijayalakshmi P (2016) Certain types of fuzzy sets in a fuzzy graph. Int J Mach Learn Cybern 7:573–579
6. Yuan Y, Wang GR, Chen L et al (2016) Efficient pattern matching on big uncertain graphs. Inf Sci 339:369–394
7. Lnokuchi A, Washio T, Motoda H (2002) An apriori-based algorithm for mining frequent substructures from graph data. Princ Data Min Knowl Discov 18:13–23
8. Inokuchi A, Washio T, Motoda H (2000) An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data, In: Proc. of the 4th European Conf. on Principles of Data Mining and Knowledge Discovery, Lyon: Springer-Verlag, pp 13–23
9. Kuramochi M, Karypis G (2001) Frequent sub-graph discovery. In: Proc. of the 2001 IEEE Int Conf. on Data Mining, San Jose: IEEE Computer Society, pp 313–320
10. Yan X, Han J (2002) GSpanGraph-based Sub-Structure Pattern Mining, In: Proc. of the 2002 IEEE Int Conf. on Data Mining, MaebashiIEEE Computer Society, pp 721–724
11. Huan J, Wang W, Prins J (2002) Efficient mining of frequent sub-graphs in the presence of isomorphism. In: Proc. of the 2003 IEEE Int Conf. on Data Mining, Melbourne: IEEE Computer Society, pp 549–552
12. Guo LX, Zhang DT, Chen L et al (2011) Research and application of data sieving algorithm based on GSpan. Appl Res Comput 28:2071–2072
13. Yan X, Han J (2003) Closegraph: mining closed frequent graph patterns. In: Proc. of the 9th ACM SIGKDD Int Conf. on Knowledge Discovery and Data Mining, Washington: ACM, pp 286–295
14. J. Huan, W. Wang, J. Prins, et al, SPIN: Mining Maximal Frequent Sub-graphs from Graph Databases, In: Proc. of the 10th ACM SIGKDD Int Conf. on Knowledge Discovery and Data Mining, Seattle: ACM, pp.581-586, 2004
15. Zou ZN, Zhu R (2013) Mining top-K maximal cliques from large uncertain graphs. Chin J Comput 36:2146–2155
16. Li MP, Gao H, Zou ZN (2014) K-reach query processing based on graph compression. J Softw 25:797–812
17. Li MP, Zou ZN, Gao H et al (2013) Computing expected shortest distance in uncertain graph. J Comput Res Dev 49:2208–2220
18. Zou Z, Li JZ, Gao H et al. (2010) Discovering Frequent Subgraph Over Uncertain Graph Database Under Probabilistic Sementids, In: Proc. of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'10), Washington, DC, USA, pp 633–642
19. Han M, Zhang W, Li JZ (2010) RAKING: an efficient K-maximal frequent pattern mining algorithm on uncertain graph database. J Comput 33:1387–1395
20. Wang WL, Li JZ (2013) MUSIC: an efficient of mining frequent sub-graph patterns in uncertain graph databases. Intell Comput Appl 3:20–23
21. Liu Y, Wang Y, Shang XQ (2014) An uncertain graph classification algorithm based on discriminative sub-graphs. Journal of Shanxi Normal University(Natural Science Edition), vol 42, pp 16–19
22. Hu J, He LB, Mao YM et al (2015) Research of improved mining frequent subgraph patterns in uncertain graph databases. Comput Eng Appl 51:112–116