Mining Strongly Correlated Sub-graph Patterns by Considering Weight and Support Constraints

Gangin Lee and Unil Yun[1]

Department of Computer Science,
Chungbuk National University, Republic of Korea
{abcnarak, yunei}@chungbuk.ac.kr

Abstract

Frequent graph mining is one of famous data mining fields that receive the most attention, and its importance has been raised continually as recent databases in the real world become more complicated. Weighted frequent graph mining is an approach for applying importance of objects in the real world to the graph mining, and numerous studies related to this have been conducted so far. However, all of the results obtained from this approach do not become actually useful information, and a significant portion of them may be meaningless ones even though they are weighted frequent sub-graph patterns. To overcome this problem, in this paper, we propose a novel method which can consider whether any sub-graph pattern has close correlation among elements in the pattern, called MSCG (Mining Strongly Correlated sub-Graph). In experimental results, we demonstrate that our MSCG outperforms a state-of-the-art method with respect to runtime and memory usage.

Keywords: *Affinity, Correlated pattern, Graph mining, Weighted frequent pattern mining*

1. Introduction

Data mining means a series of processes for finding hidden and useful information from large databases. Frequent pattern mining is one of the data mining fields which are most actively researched, and accordingly numerous techniques and methods related to this have been studied. However, as data derived from the real world have been complicated increasingly, the previous frequent pattern mining approaches have been faced with limitations since they deal with only simple databases composed of itemsets. To overcome this problem and mine complex data with graph forms, frequent graph mining methods [1, 2, 4, 5, 16] have been proposed, and thereafter advanced methods applying weight conditions [3, 6, 7, 10, 11] have been suggested to consider characteristics in the real world. Although the above methods find weighted frequent sub-graph patterns, they cannot determine how closely elements in any graph pattern are related. In this paper, to solve this issue, we propose a novel method for mining weighted frequent sub-graphs considering correlations among sub-graphs' elements, called MSCG (Mining Strongly Correlated sub-Graph), using a special and complex measure, called *weighted support affinity*. Through the method, we can obtain advantages in terms of mining performance as well as extract actually useful graph patterns from graph databases. Generally, mining sub-graphs from graph databases causes enormous overheads compared to mining itemsets from simple databases since lots of execution times are needed for graph isomorphism (NP-hard problem). However, since MSCG conducts pre-pruning operations with respect to sub-graphs with weak correlation, we can decrease the

[1] Corresponding Author, Unil Yun

number of generated patterns, thereby reducing the computation overheads. For this reason, MSCG can perform mining operations more quickly and efficiently.

The remaining parts in this paper are organized as follows. In Section 2, we introduce background related to our proposal and preliminaries, and in Section 3, details of MSCG are described. In Section 4, various experimental results show that MSCG has outstanding performance compared to a state-of-the-art method [8], and finally in Section 5, we conclude this paper.

2. Background

2.1. Related Work

In the early days of graph mining, methods based on BFS (Broad First Search) such as Apriori have been proposed, and in recent years, methods based on DFS (Depth First Search) such as pattern growth have been studied actively. Moreover, there are various approaches such as data streams [1], weight conditions [3, 6, 7, 10, 11], closed sub-graphs [1, 11, 13], maximal sub-graphs [11, 14], and so on. In [19], the authors proposed an algorithm for mining frequent sub-graph patterns from uncertain data, and in [20], an algorithm, which can find top-k maximal cliques in an uncertain graph, was suggested. As applications of graph mining, there are a variety of methods such as mining heavy sub-graphs in time-evolving networks [2], weighted substructure mining for image analysis [10], *etc*. In addition to those, the graph mining has been applied in many fields such as clustering [18], regression [12], classification [12], and so on.

FFSM, gSpan, Gaston, *etc*. are basic algorithms for mining frequent sub-graph patterns, and numerous studies for graph mining have been conducted on the basis of the algorithms. Among them, Gaston [8, 9] is a state-of-the-art algorithm and guarantees the fastest speed although it consumes more memory since the algorithm uses an additional data structure, *embedding list*. To perform mining process efficiently, Gaston divides mining steps into the three cases, a path, a free tree, and a cyclic graph, and selects appropriate mining processes depending on current situations.

In frequent itemset mining, a variety of methods [15, 17] have been studied to find patterns with close correlation. After that, algorithms for considering patterns' correlation in graph mining have been published [6, 7], where they confirm degree of correlation among elements of any sub-graph by using a special measure, called *affinity*. However, these methods consider only simple correlation in terms of a support or a weight, not complex correlation. Motivated by this problem, in this paper, we propose and apply a complex measure, *weighted support affinity*, which can consider complex and strong correlation.

2.2. Preliminaries

Note that we describe this paper based on a simple, undirected, and labeled graph form to help understand graph mining procedure and the proposed techniques and algorithm, but it is trivial to apply another graph forms, such as a multiple and directed graph structure, to our techniques or algorithm.

Every graph consists of vertices and edges. Let G be a simple, undirected, and labeled graph, V be a set of vertices in G, and E be a set of edges in G, and then V and E are defined as $V = \{v \mid v \in V\}$, $E = \{(v_1, v_2) | v_1, v_2 \in V \text{ and } v_1 \neq v_2\}$, where any edge, ($v$1, v2) is equal to ($v_2$, v_1) since all of the edges in G does not have any direction. If two certain graphs with vertices and edges satisfying the properties have the above relation, they are regarded as isomorphic

graphs. Let L be a function which returns labels of vertices and edges. Then, we denote G as $G=(V,E,L)$. Thus, given two graphs, G_1 and G_2, they are denoted as $G_1 = (V_1,E_1,L_1)$ and $G_2 = (V_2,E_2,L_2)$, and an embedding of G_1 in G_2 satisfies the following injective function, $\forall v \in V_1 -> l_1(v)=l_2(f(v))$, $\forall (v_1,v_2) \in E_1 -> (f(v_1), f(v_2)) \in E_2$ and $l_1(v_1,v_2)=l_2(f(v_1), f(v_2))$. We consider G_1 and G_2 satisfying these properties as $G_1 \subseteq G_2$. Moreover, they are isomorphic graphs if $G_1 \subseteq G_2$ is also true. Note that, in graph mining, certain graphs with the same vertices and edges (*i.e.* the isomorphic nature) in one graph transaction are considered only once. That is, although any sub-graph pattern occurs in a graph transaction many times, the support of this sub-graph is assigned as 1.

Supports of sub-graphs are calculated by considering how many they occurs in a graph database, where they are denoted as percentages from 0% to 100% or natural numbers. Weighted supports of them are additionally considered to calculate average weights of edges. That is, we can gain weighted supports of sub-graphs by multiplying their support by corresponding average weights. If any value calculated through these steps is greater than or equal to a given minimum support threshold, we call the graph with the valid value a weighted frequent sub-graph. Note that we consider edge weights in this paper since edge information can distinguish all of the sub-graphs uniquely, but vertices cannot do that. The details will be presented in the next section.

3. MSCG: Mining Strongly Correlated sub-Graph patterns

In this section, we describe the proposed measure, *weighted support affinity* and present how the measure and its characteristics are applied to weighted frequent graph mining steps. We also analyze pruning effect by using our technique and provide details of MSCG method.

3.1. Strongly Correlated Sub-graph

A sub-graph pattern consists of vertices and edges, where each element includes support and weight information. In this paper, we use edge's support and weight to distinguish sub-graph patterns.

Property 1. Among the elements composing graphs, a set of edges only becomes a standard for distinguishing each graph.

Proof. Given a path, *P* and a cyclic graph, *C*, they can have the same vertex information. For example, a path {D-a-A-c-D} and a cyclic graph {D-a-A-c-D-b} exist in figure 1, and they have the same set of vertices, {D, A, C}. Therefore, we cannot distinguish them with the only vertex information. However, in case we use edge information for the distinction, their sets of edges are {a, c} and {a, c, b} respectively, and thus they can be classified. ■

In this paper, we consider *weighted support affinity* to find sub-graphs with strong correlation among elements in any sub-graph pattern, and this measure is computed as follows.

Definition 1. Given a sub-graph, *SG*, a set of edges, $E = \{e_1, e_2, e_3, \ldots, e_n\}$, and a set of weights corresponding to E, $W = \{w_1, w_2, w_3, \ldots, w_n\}$, then a set of weighted supports for edges, *WS* is denoted as $WS = \{ws_1, ws_2, ws_3, \ldots, ws_n\}$, where $ws_n = w_n * support(e_n)$. Then, a weighted support affinity for *SG*, *WSA(SG)* is calculated as shown in the following formula.

$$WSA(SG) = min_{1 \le i \le n}(ws_i) / max_{1 \le i \le n}(ws_i) \quad (1)$$

In the above formula, elements in *SG* are less related to each other if *WSA(SG)* is close to 0 while they are more related with each other if *WSA(SG)* is near to 1. To determine whether

any sub-graph becomes valid or not, we use a threshold, *minimum weighted support affinity*, and this has a real number value between 0 and 1.

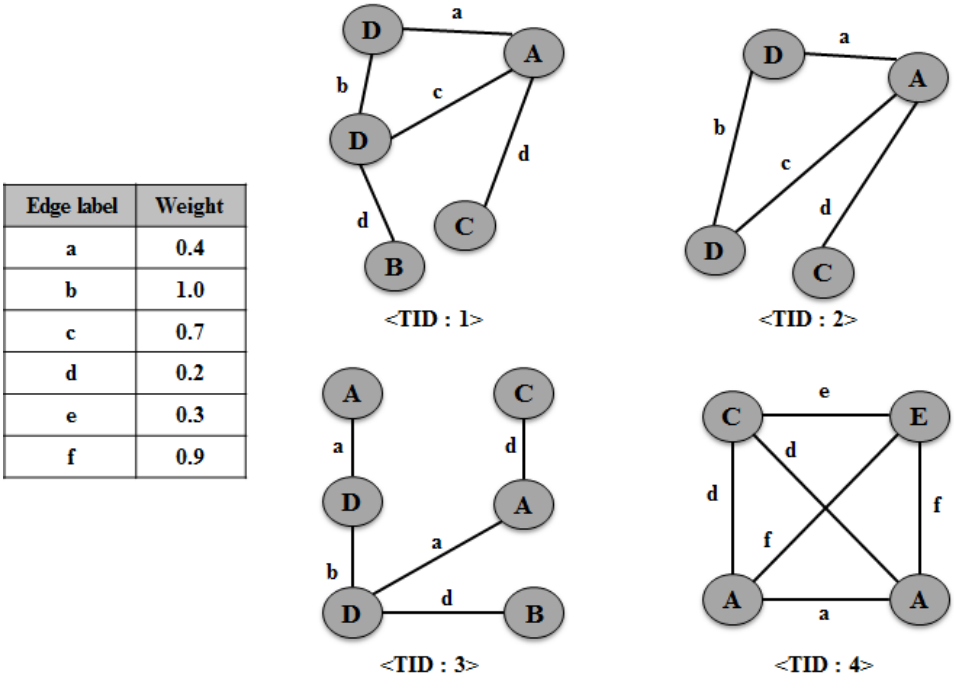

Figure 1. An Example of Graph Database

3.2. Pruning Strategy for Weakly Correlated Sub-graphs

By using the measure and its characteristics from Definition 1, we can efficiently reduce search space for mining sub-graphs by performing pre-pruning operations for invalid patterns as well as find strongly correlated sub-graph patterns effectively, and thereby MSCG guarantees faster execution time and more compact memory usage.

Pre-pruning condition 1. Depending on basic property of weighted frequent pattern mining, any sub-graph and its possible extensions are wholly eliminated if multiplying a support of the pattern by *MaxW* is lower than *minimum threshold*, where *MaxW* is set as the maximum weight which the current pattern and all of its super patterns can have.

Pre-pruning condition 2. If *WSA* value of a certain sub-graph is less than *minimum weighted support affinity*, this sub-graph and its expanded graphs are also pruned.

Pruning condition 1. If a weighted support of any sub-graph, *WSUP* is less than *minimum threshold* even though the graph does not satisfy the above two conditions, this becomes a useless pattern and is not extracted, where *WSUP* is to multiply pattern's support by its average edge weight.

Therefore, sub-graphs that do not satisfy all of the pruning conditions are only considered as valid ones and mined ultimately.

Lemma 1. The proposed measure, *WSA* satisfies *Anti-monotone* property, and thus certain sub-graphs pre-pruned by *WSA* are exactly useless patterns and any loss does not occur in mining process.

Proof. *Anti-monotone* property means that if any pattern is infrequent, all of the super patterns of the pattern are also infrequent. Let *SG* be a sub-graph, *e* be an extended edge, *SG'* be a graph extended by *e*, and λ be *minimum weighted support affinity*. Then, we can consider a relation between *WSA(SG)* and *WSA(SG')* as the three cases. In case $ws_e > max_{1 \le i \le n}(ws_i)$ *for* $ws_i \in SG$, ws_e is assigned as a new maximum value and thus $WSA(SG') = min_{1 \le i \le n}(ws_i) / ws_e$, where $WSA(SG) > WSA(SG')$ since the denominator becomes larger. In case $min_{1 \le i \le n}(ws_i) > ws_e$, the minimum value is replaced with ws_e and therefore, $WSA(SG') = ws_e / max_{1 \le i \le n}(ws_i)$. Then, $WSA(SG) > WSA(SG')$ since the numerator becomes smaller. In the last case, if ws_e is between *min* and *max*, $WSA(SG) = WSA(SG')$ since there is no effect for the *WSA* value. Considering these cases, we can derive a result, $WSA(SG) \ge WSA(SG')$. As a result, if *SG* is a meaningless pattern, i.e. $WSA(SG) < \lambda$, *SG'* is also invalid one according to the above result since $WSA(SG') \le WSA(SG) < \lambda$. That is, *WSA* satisfies *Anti-monotone* property. Consequently, it is certain that the pruning technique by *WSA* does not generate any loss or error. ■

Example 1. Assume that *minimum threshold*, δ is 2 and *minimum weighted support affinity*, λ is 0.5. Then, in the Figure 1, a path {D-b-D-a-A} has a support, 3 and an average weight, 0.7, and therefore, the path satisfies δ(3*0.7=2.1). However, that pattern becomes an invalid one in the end since *WSA*({D-b-D-a-A}) is 0.4 which is no greater than λ.

input: a graph database, *GDB*, a minimum threshold δ, a minimum weighted support affinity, λ
output: a set of strongly correlated sub-graphs, *S*

Mining_sub-graph(GDB, δ, λ, ω)
1. find all vertices and edges such that support * *MaxW* ≥ δ in *GDB*
2. for each vertex, *v* in a set of found vertices, *V* do
3. a sub-graph, *G* ← *v*
4. a set of valid edge, *E'* ← edges attached to *v* among the weighted frequent edges
5. a pattern state, *PS* ← "*path*"
6. *S* = *S* ∪ *Extending_graph(G, E', PS)*
7. return *S*

Extending_graph(a sub-graph G, a set of edges E, a pattern state PS)
1. for each edge, *e* in *E* do
2. if *PS* is "*path*" or "*free tree*" do
3. generate an extended path or free tree, *G'* of *G* adding *e* and corresponding vertex, *v*
4. else generate an extended cyclic graph, *G'* of *G* adding only *e*
5. compute *WSA(G')* and *PS* ← the current state of *G'*
6. if *WSA(G')* ≥ λ and *WSUP(G')* ≥ δ do
7. *S* = *S* ∪ *G'*
8. else *e* ← the next edge in *E* and goto line 1
9. *E'* ← a set of valid edges that can be attached to *G'*
10. *S* = *S* ∪ *Extending_graph(G, E', PS)*
11. return *S*

Figure 2. MSCG Algorithm

Thus, an extended path {D-b-D-a-A-d-C} is also pruned by the Lemma 1, where this pattern's *WSA* is 0.2(=0.2/1.0) and therefore the pattern becomes a meaningless sub-graph actually.

3.3. MSCG Algorithm

Figure 2 represents MSCG algorithm applying the measure defined from Section 3.1 and the pruning strategy in Section 3.2. Mining procedure of MSCG(*Mining_sub-graph*) is as follows. MSCG first finds frequent vertices and valid edges such that edges support * *MaxW* ≥ δ(line 1), and then conducts graph extension for each vertex and edge(line 2~6). The function, *Extending_graph* is performed as the following steps. The function generates an extended graph, *G'* depending on the current pattern's state as adding edges one by one (line 2~4). Thereafter, it calculates *WSA(G')* and confirms the current state for *G'*(line 5). If *G'* satisfies the conditions at line 6, this pattern becomes a meaningful sub-graph and is stored into *S*(line 7). In contrast, if those conditions are not satisfied, the function discards the current extended edge, *e* and performs the next extension with the following edge again (line 8). If *G'* is valid, the function extracts a set of expansible edges with respect to *G'* and then calls itself recursively (line 9~10). After terminating all of the mining operations, we can obtain strongly correlated sub-graph patterns from the inputted graph database.

4. Analysis of Experimental Results

We compare the proposed algorithm, MSCG with a state-of-the-art algorithm, Gaston [8, 9] in this section, where those algorithms were written as C++ language and ran with 3.33GHz CPU, 3GB RAM, and WINDOWS 7 OS. In experiments, two real graph datasets, DTP and PTE datasets are used, and the details of these datasets were introduced in [9]. Each algorithm is evaluated in terms of runtime and memory usage, and edge weights for each dataset are set as random values between 0.5 and 0.8.

4.1. Runtime Analysis

Figure 3 and 4 show runtime results for each algorithm regarding the two datasets.

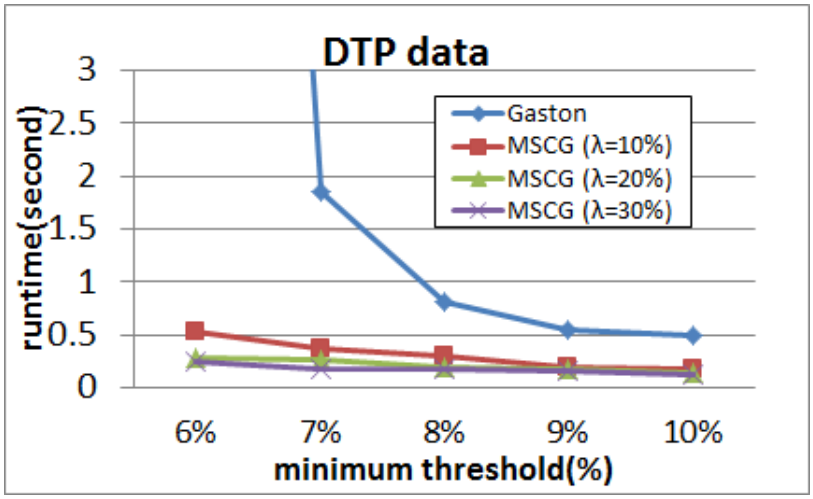

Figure 3. Runtime Results for DTP Dataset

When *minimum weighted support affinity*, λ is assigned as 30% in the DTP dataset, it is observed that MSCG guarantees the fastest runtime in every case as shown in the figure 3. In addition, MSCGs with the other λ values also represent higher speed than that of Gaston. When *minimum threshold*, δ is 9%~10%, Gaston presents relatively favorable runtime, but its performance is sharply worsened as δ becomes lower. The gap between the algorithms represents how quickly MSCG performs mining operations through *WSA*. In the PTE dataset, MSCG also has the best results regardless of λ values although the gap is small when δ is relatively high as shown in the Figure 4. The higher λ is, the more decreased the number of extracted sub-graphs is, and the extracted ones have much more strong correlation.

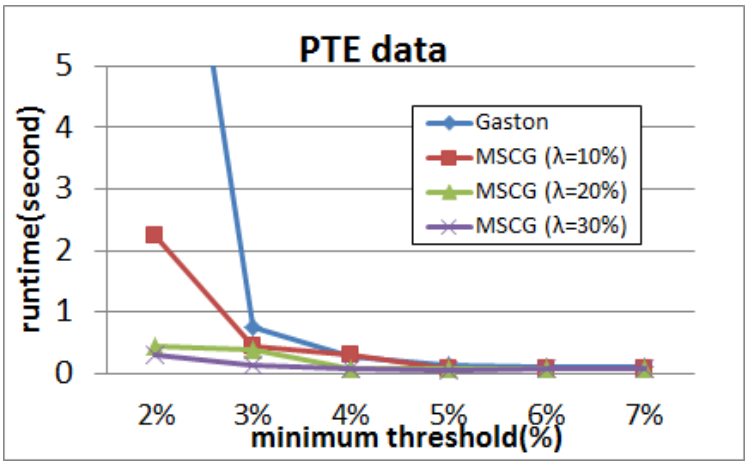

Figure 4. Runtime Results for PTE Dataset

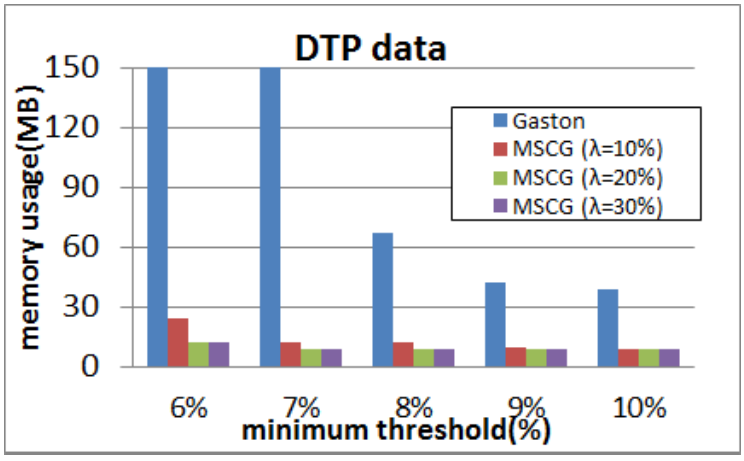

Figure 5. Memory Usage Results for DTP Dataset

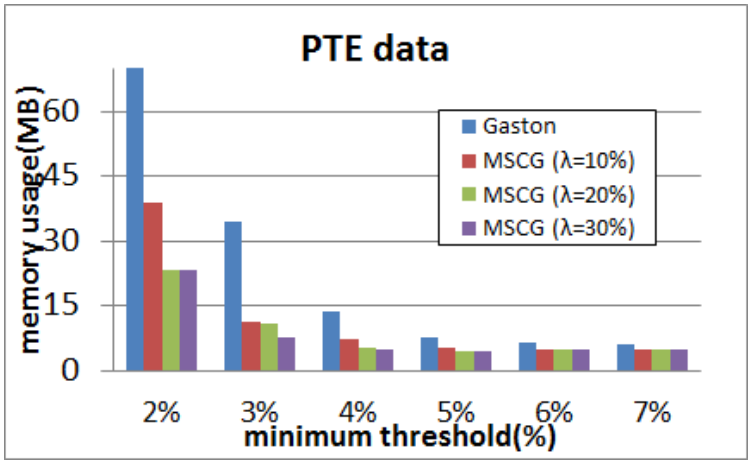

Figure 6. Memory Usage Results for PTE Dataset

4.1. Memory Usage Analysis

Figure 5 and 6 illustrate memory usage results for each algorithm, where MSCG shows the most efficient memory usage in DTP and PTE datasets regardless of λ in common with the runtime test. Especially, it is observed that memory consumption of MSCG is decreased as λ becomes higher. The result of DTP dataset has more large gaps compared with that of PTE dataset as shown in the figures. The reason is as follows. DTP has more graph transactions and more complex structures, and therefore DTP generally mines sub-graphs more than PTE. Although the obtained results are large, it does not mean that strongly correlated patterns included in the results are also large. Since many sub-graphs from DTP have weak correlation in the actual mining result, a great portion of them is pruned by the *WSA* measure, and thus we can observe that MSCG guarantees obvious performance improvements as shown in the figures.

5. Conclusions

In this paper, we proposed a measure, called *weighted support affinity*, to find strongly correlated sub-graphs and a mining algorithm, named MSCG, applying *weighted support affinity* and its property. MSCG not only mined meaningful patterns with strong correlation among the elements in sub-graphs efficiently but also advanced performance as reducing search space by the proposed measure. A variety of experiments presented that our MSCG outperforms the previous method, Gaston in all of the cases. MSCG's mining process was conducted in static graph database environment, but if MSCG's techniques are applied to data stream, high utility, closed, and maximal graph mining approaches, we expect that our method will have a significant effect on their mining performance and efficiency.

Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF No. 2012-0003740 and 2012-0000478).

References

[1] A. Bifet, G. Holmes, B. Pfahringer and R. Gavalda, "Mining Frequent Closed Graphs on Evolving Data Streams", Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2011)** August 21-24; San Diego, CA, USA.
[2] P. Bogdanov, M. Mongiovi and A.K. Singh, "Mining Heavy Subgraphs in Time-Evolving Networks", Proceedings of the IEEE 11th International Conference on Data Mining, **(2011)** December 11-14; Vancouver, BC, Canada.
[3] S. Günnemann and T. Seidl, "Subgraph Mining on Directed and Weighted Graphs", Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, **(2010)** June 21-24; Hyderabad, India.
[4] P. Hintsanen and H. Toivonen, "Finding reliable subgraphs from large probabilistic graphs", Data Mining Knowledge Discovery, vol. 17, no. 1, **(2008)**.
[5] Y. Jia, J. Zhang and J. Huan, "An efficient graph-mining method for complicated and noisy data with real-world applications", Knowledge Information Systems, vol. 28, no. 2, **(2011)**.
[6] C. Jiang, F. Coenen and M. Zito, "Frequent Sub-graph Mining on Edge Weighted Graphs", Proceedings of the 12th international conference on Data warehousing and knowledge discovery, **(2010)** August 30-September 2; Bilbao, Spain.
[7] G. Lee and U. Yun, "An Efficient Approach for Mining Frequent Sub-graphs with Support Affinities", Proceedings of the 6th international conference on Convergence and Hybrid Information Technology, **(2012)** August 23-25; Daejeon, Korea.
[8] S. Nijssen and J. N. Kok, "The Gaston Tool for Frequent Subgraph Mining", Electronic Notes in Theoretical Computer Science, vol. 127, no. 1, **(2005)**.
[9] S. Nijssen and J. N. Kok, "A quickstart in frequent structure mining can make a difference", Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2004)** August 22-25; Seattle, Washington, USA.
[10] S. Nowozin, K. Tsuda, T. Uno, T. Kudo and G.H. Bakir, "Weighted Substructure Mining for Image Analysis", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2007)** June 18-23; Minneapolis, Minnesota, USA.
[11] T. Ozaki and M. Etoh, "Closed and Maximal Subgraph Mining in Internally and Externally Weighted Graph Databases", 25th IEEE International Conference on Advanced Information Networking and Applications Workshops, **(2011)** March 22-25; Biopolis, Singapore.
[12] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo and K. Tsuda, "gBoost: a mathematical programming approach to graph classification and regression", Machine Learning, vol. 75, no. 1, **(2009)**.
[13] I. Takigawa and H. Mamitsuka, "Efficiently mining δ-tolerance closed frequent subgraphs", Machine Learning, vol. 82, no. 2, **(2011)**.
[14] L. T. Thomas, S. R. Valluri and K. Karlapalem, "MARGIN: Maximal frequent subgraph mining", Transactions on Knowledge Discovery from Data, vol. 4, no. 3, **(2010)**.
[15] H. Xiong, P. N. Tan and V. Kumar, "Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution", Proceedings of the 3rd IEEE International Conference on Data Mining, **(2003)** November 19-22; Melbourne, Florida, USA.
[16] X. Yan, H. Cheng, J. Han and P. S. Yu, "Mining significant graph patterns by leap search", Proceedings of the ACM SIGMOD International Conference on Management of Data, **(2008)** June 10-12; Vancouver, BC, Canada.
[17] U. Yun, "WIS: Weighted Interesting Sequential Pattern Mining with a Similar Level of Support and/or Weight", ETRI Journal, vol. 29, no. 3, **(2007)**.
[18] Y. Zhou, H. Cheng and J. X. Yu, "Clustering Large Attributed Graphs: An Efficient Incremental Approach", Proceedings of the IEEE 10th International Conference on Data Mining, **(2010)** December 14-17; Sydney, Australia.
[19] Z. Zou, J. Li, H. Gao and S. Zhang, "Mining Frequent Subgraph Patterns from Uncertain Graph Data", IEEE Transactions on Knowledge Data Engineering, vol. 22, no. 9, **(2010)**.
[20] Z. Zou, J. Li, H. Gao and S. Zhang, "Finding top-k maximal cliques in an uncertain graph", Proceedings of the 26th International Conference on Data Engineering, **(2010)** March 1-6; Long Beach, California, USA.

Authors

Gangin Lee

Gangin Lee received the BS degree in Computer Engineering from Chungbuk National University, Chungbuk, Korea, in 2012. He is currently working toward the MS degree in Department of Computer Science, Chungbuk National University. His current research interests include data mining, information retrieval, and database systems.

Unil Yun

Unil Yun received the MS degree in Computer Science from Korea University, Seoul, Korea, and the PhD degree in Computer Science from Texas A&M University, Texas, USA. He is currently a professor in Department of Computer Science at Chungbuk National University, Chungbuk, Korea. His current research interests include data mining, information retrieval, and database systems.