Mining Discriminative Patterns from Graph Structured Data with Constrained Search

Kiyoto Takabayashi[1], Phu Chien Nguyen[1], Kouzou Ohara[1], Hiroshi Motoda[2], and Takashi Washio[1]

[1] I.S.I.R., Osaka University,
8-1, Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
{kiyoto_ra,chien,ohara,washio}@ar.sanken.osaka-u.ac.jp
[2] AFOSR/AOARD
7-23-17, Roppongi, Minato-ku, Tokyo 106-0032, Japan
hiroshi.motoda@aoard.af.mil

Abstract. A graph mining method, Chunkingless Graph-Based Induction (Cl-GBI), finds typical patterns that appear in graph structured data by the operation called chunkingless pairwise expansion which generates pseudo-nodes from selected pairs of nodes in the data. Cl-GBI enables to extract overlapping subgraphs, while it requires more time and space complexities. Thus, it happens that Cl-GBI cannot extract patterns that need be large enough to describe characteristics of data within a limited time and a given computational resource. In such a case, extracted patterns may not be so much of interest for domain experts. To mine more discriminative patterns which cannot be extracted by the current Cl-GBI, we introduce a search algorithm guided by domain knowledge or interests of domain experts. We further experimentally show that the proposed method can efficiently extract more discriminative patterns using both synthetic and real world datasets.

1 Introduction

Over the last decade, there has been much research work on data mining which intends to find useful and interesting knowledge from massive data. A number of studies have been made in recent years especially on mining frequent patterns from graph structured data, or simply graph mining because of the high expressive power of graph representation [1, 13, 6, 12, 4, 5].

Chunkingless Graph Based Induction (Cl-GBI) [8] is an extension of Graph Based Induction (GBI) [13] that can extract typical patterns from graph structured data by stepwise pair expansion, i.e., by recursively chunking two adjoining nodes. Similarly to GBI, Cl-GBI adopts the stepwise pair expansion principle, but never chunks adjoining nodes and contracts the graph. Instead, Cl-GBI regards a pair of nodes as a *pseudo node* and assigns a new label to it. This operation can fully solve the reported problems caused by chunking, i.e., ambiguity in selecting nodes to chunk and incompleteness of the search. However Cl-GBI requires more time and space compelexities to gain the ability of extracting overlapping patterns. Thus, it happens that Cl-GBI cannot extract patterns

that need be large enough to describe characteristics of data within time and space limitation. In such a case, extracted patterns may not be so much of interest for domain experts.

To improve the search efficiency, in this paper, we propose a method of guiding the search of Cl-GBI using domain knowledge or interests of domain experts. The basic idea is adopting patterns representing knowledge or interests of domain experts as constraints on the search, in order to effectively restrict the search space and extract more discriminative or interesting patterns than those which can be extracted by the current Cl-GBI. We also experimentally show the effectiveness of the proposed search method by applying the constrained Cl-GBI to a synthetic dataset and the hepatitis dataset which is a real world dataset.

In this paper, we deal with only connected labeled graphs, and use information gain [10] as the discriminativity criterion. In what follows, "a pair" denotes a pair of adjoining nodes in a graph.

2 Constrained Search for Cl-GBI

2.1 Chunkingless Graph-Based Induction(Cl-GBI)

Stepwise pair expansion is an essential operation in GBI, which recursively generates new nodes from pairs of two adjoining nodes selected according to a certain criterion based on frequency, and replaces all of its occurrences in graphs with a node having a newly assigned label. Namely each graph is rewritten each time a pair is chunked, and never restored in any subsequent chunking [3]. Although thanks to this chunking mechanism, GBI can efficiently extract patterns from either a huge single graph or a set of graphs, it involves ambiguity in selecting nodes to chunk, which causes a crucial problem, i.e., possibility of overlooking some overlapping subgraphs due to inappropriate chunking order. Beam search adopted by Beam-wise GBI(B-GBI) [6] can alleviate this problem by chunking the b (beam width) most frequent pairs and copying each graph into respective states, but not completely solve it because chunking process is still involved.

In contrast to GBI and B-GBI, Cl-GBI does not chunk a selected pair, but regards it as a *pseudo node* and assigns a new label to it. Thus, graphs are not "compressed" nor copied over the iterative *pseudo-chunking* process. We refer to each iteration in Cl-GBI as "level". The algorithm of Cl-GBI is shown in Fig. 1. The search of Cl-GBI is controlled by the following parameters: a beam width b, the maximal number of levels of pseudo-chunking N, and a frequency threshold θ. In other words, at each level, the b most frequent pairs are selected from a set of pairs whose frequencies are not less than θ, and are pseudo-chunked.

2.2 Patterns Used as Constraints

The current Cl-GBI blindly extracts a huge number of frequent pairs without any clues other than frequency. However, if the goal is finding patterns which are

[3] This does not mean that the link information of the original graphs is lost. It is always possible to restore how each node is connected in the extracted subgraphs.

Input. A graph database D, a beam width b, the maximal number of levels of pseudo-chunking N, a frequency threshold θ

Output. A set of typical patterns S

Step 1. Extract all the pairs consisting of two connected nodes in the graphs, register their positions using node id (identifier) sets. From the 2nd level on, extract all the pairs consisting of two connected nodes with at least one node being a new pseudo-node.

Step 2. Count frequencies of extracted pairs and eliminate pairs whose frequencies count below θ.

Step 3. Select the b most frequent pairs from among the remaining pairs at Step 2 (from the 2nd level on, from among the unselected pairs in the previous levels and the newly extracted pairs). Each of the b selected pairs is registered as a new node. If either or both nodes of the selected pair are not original but pseudo-nodes, they are restored to the original patterns before registration.

Step 4. Assign a new label to each pair selected at Step 3 but do not rewrite the graphs. Go back to Step 1.

Fig. 1. Algorithm of Cl-GBI

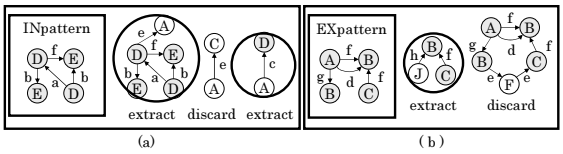

Fig. 2. Examples of INpatterns and EXpatterns

either discriminative or of interest for domain experts, the current method is too naive and inefficient in both time and space complexities. For example, when we analyzed the hepatitis dataset [11] provided by Chiba University Hospital with Cl-GBI, domain experts (medical doctors) expected that patterns interesting for them were extracted, but in fact we could not find satisfactory ones.

Therefore, in this paper, we introduce domain knowledge or interests of domain experts and impose them as constraints on patterns extracted in Cl-GBI. We represent such domain knowledge and interests as graphs and call them the *constraint patterns*. Although various types of constraint patterns can be considered, in this paper, we focus on the following two types of patterns: patterns to be included in extracted patterns and patterns not to be included in them. We refer to them as *INpatterns* and *EXpatterns*, respectively, and define the following two types of constraints: "*extracted patterns must include INpatterns*" (Constraint 1) and "*extracted patterns must not include EXpatterns*" (Constraint 2).

Figures 2 (a) and (b) show the examples of Constraints 1 and 2, respectively. Note that in case of Constraint 1, not only patterns including the given INpatterns, but also patterns including at least one of their proper subgraphs should be extracted in order not to prevent patterns satisfying the imposed constraints from being generated in the succeeding steps based on the stepwise pair expansion principle. The case is illustrated at the far right in Fig. 2 (a). In addition, in case of Constraint 1, we can discard a pair if it does not include any node/link labels appearing in the given INpatterns as shown in Fig. 2 (a) because such a pair can never grow to a pattern that includes at least one of the given INpatterns.

2.3 Design of Constrained Search

To guide the search process of Cl-GBI using INpatterns/EXpatterns, we have to check if an enumerated pair includes them, which requires additional subgraph isomorphism checking known to be NP-complete [2]. Therefore, to reduce the computational cost, we should detect pairs that have no possibility of including constraint patterns before the checking. For that detection, we define two conditions based on the number of node/link labels in a pattern.

First of all, for two arbitrary pairs, or patterns x and y, we define a quantity T_{num} as follows:

$$T_{num}(x, y) = \sum_{L_k \in L(y)} f(x, L_k), \tag{1}$$

where $L(y)$ is a set of labels appearing in y, and $f(x, L_k)$ is the number of occurrences of the label $L_k \in L(y)$ in x. Note that if x is identical to y, $T_{num}(x, y)$ must be equal to $T_{num}(y, y)$. Similarly, $T_{num}(x, y)$ must be greater than $T_{num}(y, y)$ if y is a subgraph of x. Consequently, given a constraint pattern T_j, we can skip subgraph isomorphism checking for an enumerated pattern P_i if $T_{num}(P_i, T_j) < T_{num}(T_j, T_j)$ because P_i never includes T_j.

Furthermore, it is noted that in order for P_i to include T_j, for every label appearing in T_j, the number of its occurrences in P_i has to be greater than or equal to that in T_j. Namely, we can skip subgraph isomorphism checking for P_i if P_i does not satisfy this condition. To check this condition, we define the following boolean value P_{info} for two patterns x and y.

$$P_{info}(x, y) = \bigwedge_{L_k \in L(y)} p(x, y, L_k), \tag{2}$$

where

$$p(x, y, L_k) = \begin{cases} true & \text{if } f(x, L_k) \geq f(y, L_k), \\ false & \text{otherwise.} \end{cases}$$

If $P_{info}(P_i, T_j)$ is *true*, then subgraph isomorphism checking has to be done; otherwise it can be skipped.

These ideas are summarized in Fig. 3 as the algorithm, which is invoked in the algorithm shown in Fig. 1 and provides a set of candidate pairs to be pseudo-chunked at each level. $PD(P_i, T_j)$ in Fig. 3 is the procedure for subgraph isomorphism checking, which returns true if P_i includes T_j; otherwise false.

3 Experimental Evaluation

To evaluate the proposed method, we implemented Cl-GBI with the algorithm shown in Fig. 3 on PC (CPU: Pentium 4 3.2GHz, Memory: 4GB, OS: Fedora Core release 3) in C++, and applied this *constrained Cl-GBI* to both synthetic and real-world datasets consisting of directed graphs. The current system has a limitation that either INpattern constraints or EXpattern constraints can be

```
ExtPair(D, T, L, L_v, M)
Input: a database D, a set of constraint patterns T, the current level L_v,
       a set of extracted pairs L (initially empty),
       the constraint mode M (either "INpattern" or "EXpattern");
Output: a set of extracted pairs L with newly extracted pairs;
begin
  if L_v = 1 then
    if M = "INpattern" then
      Enumerate pairs in D, which consist of nodes or links
      appearing in T, and store them in E;
    else
      Enumerate all the pairs in D and store them in E;
  else
    Enumerate pairs, which consist of one or both
    pseudo nodes in L, and store them in E;
  for each P_i ∈ E begin
    if P_i is marked then L := L ∪ {P_i}; next;
    else register := 1;
    for each T_j ∈ T begin
      if T_num(P_i, T_j) ≥ T_num(T_j, T_j) then
        if P_info(P_i, T_j) = true then
          if M = "INpattern" then
            if PD(P_i, T_j) = true then mark P_i;
          else
            if PD(P_i, T_j) = true then
              discard P_i; register := 0; break;
    end
    if register = 1 then L := L ∪ {P_i};
  end
  return L;
end
```

Fig. 3. Algorithm of the constrained pattern extraction

imposed at a time. We verified the efficiency of the proposed method with the synthetic dataset, and also confirmed that it could extract patterns which are more discriminative than those by the current Cl-GBI with the real-world one.

3.1 Synthetic Dataset

Experimental Settings: The synthetic dataset was generated in a random manner same as [9], and divided into two classes of equal size, "active" and "inactive". Then, as discriminative patterns, we generated 4 kinds of subgraphs, or "basic patterns" shown in Fig. 4, and embedded them in transactions of the class "active". In Fig. 4, "f" and "IG" denote frequency and information gain, respectively. The statistics on the size of resulting graphs are shown in Table 1.

In this experiment, we used as INpatterns a subgraph of each basic pattern shown in Fig. 4 separately to extract the corresponding basic pattern, and set the parameters of Cl-GBI as follows: $b = 5, 7, 10$,$N = 10$, and $\theta = 0\%$.

Experimental Results: We observed the computation time and the level at which the basic pattern was extracted by the constrained Cl-GBI in each case. The results are summarized in Table 2, in which t and level denote the observed computation time and level, respectively, and the values in the parentheses are

Table 1. Size of graphs of the synthetic dataset and the hepatitis dataset

class	Synthetic		Hepatitis	
	active	inactive	R	N
number of graphs	150	150	38	56
average number of nodes in a graph	50	50	104	112
total number of nodes	7,524	7,502	3,944	6,296
kinds of node labels	25		12	
average number of links in a graph	498	495	108	117
total number of links	74,631	74,198	4,090	6,577
kinds of link labels	25		30	

Table 2. Experimental results for the synthetic dataset

constraint pattern	$b = 5$			$b = 7$			$b = 10$		
	t[sec]	level	IG	t[sec]	level	IG	t[sec]	level	IG
a2	119(5976)	4(10)	0.0421	133(6537)	4(8)	0.0421	170(6810)	4(5)	0.0603
b2	42(–)	4(–)	0.0421	32(–)	3(–)	0.0421	42(–)	3(–)	0.0603
c2	168(–)	6(–)	0.0421	166(–)	5(–)	0.0421	150(–)	4(–)	0.0603
d2	167(–)	6(–)	0.0421	92(–)	4(–)	0.0421	75(–)	3(–)	0.0603

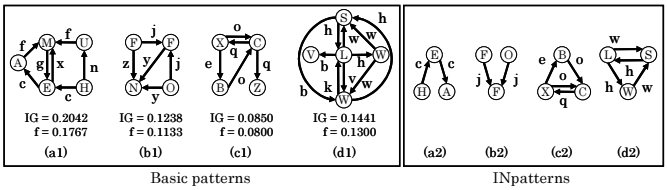

Fig. 4. Basic patterns and INpatterns used in the experiments for the synthetic dataset

corresponding results by the current Cl-GBI. "–" represents that the current Cl-GBI could not extract the basic pattern. The column "IG" in Table 2 denotes the maximal information gain achieved by the current Cl-GBI at the same level with the same parameter settings.

From this table, we can see that the constrained Cl-GBI succeeded in extracting the basic pattern in all the cases at an early stage, while the current Cl-GBI extracted only patterns having considerably less information gain and could not extract some of basic patterns. From these results, it is said that the constrained Cl-GBI can extract patterns which are more discriminative than those by the current Cl-GBI in a less computation time and smaller computational resources.

3.2 Real-world Dataset

Experimental Settings: As the real-world dataset, we used the two classes in the hepatitis dataset, *Response* and *Non-Response*, denoted by R and N, respectively [3]. R consists of patients to whom the interferon therapy was effective, while N consists of those to whom it was not effective. We converted the records of each patient into a graph in the same way as [3]. The statistics on the size of resulting graphs are shown in Table 1.

In this experiment, we used 4 sets of INpatterns shown in Fig. 5, in which (a) to (c) represent typical examination results for patients belonging to R [7],

Table 3. Experimental results for the hepatitis dataset

	time[sec]	max information gain
original	44,973	0.1139 (L:4, t:1292)
No.1	9,355	0.1076 (L:3, t:18)
No.2	6,893	0.1698 (L:5, t:376)
No.3	20,495	0.1110 (L:3, t:55)
No.4	4,970	0.1297 (L:4, t:39)

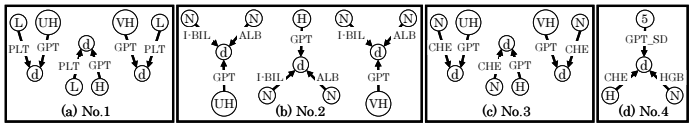

Fig. 5. INpatterns used in the experiments for the hepatitis dataset

while (d) is the most discriminative pattern extracted by the current Cl-GBI. We refer to the pattern which is the most discriminative one among extracted patterns as the *MDpattern*. The node with the label "d" in Fig. 5 represents a certain point of time. For example, the leftmost pattern in Fig. 5 (a) means that at a certain point of time, the value of GPT (glutamic-pyruvic transaminase) is High and the value of PLT (platelet) is Low. The parameters of Cl-GBI were set as follows: $b = 10$, $N = 10$, and $\theta = 0\%$.

Experimental Results: We observed the computation time and information gain of the MDpattern in each case. The results are shown in Table 3, in which the row "original" contains the results by the current Cl-GBI with the same parameter settings. Namely, the MDpattern shown in Fig. 5 (d) is identical to that in the case of "original". "L" and "t" in parentheses denote the level and time[sec] spent to extract the MDpattern, respectively.

From Table 3, it is found that the MDpatterns extracted by the constrained Cl-GBI are more discriminative than the MDpattern by the current Cl-GBI in the cases of No.2 and No.4. In addition, the computation times in all the 4 cases using INpatterns are much less than in the case of the current Cl-GBI. From these results, we can say that given appropriate constraints, the constrained Cl-GBI could efficiently extract patterns which are more discriminative than those by the current Cl-GBI. In addition, note that the INpattern used in the case of No.4 which is the MDpattern obtained by the current Cl-GBI works as a good constraint. From this result, it is expected that running the constrained Cl-GBI repeatedly with a small L using the MDpattern extracted by the previous run as the new INpattern might allow us to extract discriminative patterns in a less computation time. Verifying this expectation is one of our future work.

4 Conclusion

In this paper, we proposed a constrained search method that effectively restricts the search space of Cl-GBI by imposing domain knowledge or interests of domain experts as constraints on patterns to be searched, and embedded it in Cl-GBI,

resulting in the constrained Cl-GBI. Experimental results showed that given appropriate constraints, the constrained Cl-GBI can extract more discriminative patterns in a less computation time than the current Cl-GBI. In addition, the results also showed the possibility that discriminative patterns extracted in earlier steps in the search may work as good constraints in the constrained Cl-GBI.

As future work, we plan to further evaluate the constrained Cl-GBI by comparing it with other graph mining methods including ones based on Inductive Logic Programming, and to verify the resulting patterns cooperating with domain experts such as medical doctors.

References

1. Cook, D. J. and Holder, L. B.: Substructure Discovery Using Minimum Description Length and Background Knowledge. Artificial Intelligence Research, Vol. 1, pp. 231–255, (1994).
2. Fortin, S.: The Graph Isomorphism Problem. Technical Report TR96-20, Department of Computer Science, University of Alberta, (1996).
3. Geamsakul, W., Yoshida, T., Ohara, K., Motoda, H., Yokoi, H., and Takabayashi, K.: Constructing a Decision Tree for Graph-Structured Data and its Applications. Fundamenta Informaticae Vol. 66, No.1-2, pp. 131–160, (2005).
4. Inokuchi, A., Washio, T., and Motoda, H.: Complete Mining of Frequent Patterns from Graphs: Mining Graph Data. Machine Learning, Vol. 50, No. 3, pp. 321–354, (2003).
5. Kuramochi, M. and Karypis, G.: An Efficient Algorithm for Discovering Frequent Subgraphs. IEEE Trans. Knowledge and Data Engineering, Vol. 16, No. 9, pp. 1038–1051, (2004).
6. Matsuda, T., Motoda, H., Yoshida, T., and Washio, T.: Mining Patterns from Structured Data by Beam-wise Graph-Based Induction. Proc. of DS 2002, pp. 422–429, (2002).
7. Motoyama, S., Ichise, R., and Numao, M.: Knowledge Discovery from Inconstant Time Series Data. JSAI Technical Report, SIG-KBS-A405, pp. 27–32, in Japanese, (2005).
8. Nguyen, P. C., Ohara, K., Motoda, H., and Washio, T.: Cl-GBI: A Novel Approach for Extracting Typical Patterns from Graph-Structured Data. Proc. of PAKDD 2005, pp. 639–649, (2005).
9. Nguyen, P. C., Ohara, K., Mogi, A., Motoda, H., and Washio, T.: Constructing Decision Trees for Graph-Structured Data by Chunkingless Graph-Based Induction. Proc. of PAKDD 2006, pp. 390–399, (2006).
10. Quinlan, J. R.: Induction of decision trees. Machine Learning, Vol. 1, pp. 81–106, (1986).
11. Sato, Y., Hatazawa, M., Ohsaki, M., Yokoi, H., and Yamaguchi, T.: A Rule Discovery Support System in Chronic Hepatitis Datasets. First International Conference on Global Research and Education (Inter Academia 2002), pp. 140–143, (2002).
12. Yan, X. and Han, J.: gSpan: Graph-Based Structure Pattern Mining. Proc. of ICDM 2002, pp. 721–724, (2002).
13. Yoshida, K. and Motoda, H.: CLIP: Concept Learning from Inference Patterns. Artificial Intelligence, Vol. 75, No. 1, pp. 63–92, (1995).