Representative Pattern Mining in Graph Collections

Niusvel Acosta-Mendoza[1,2], Jesús Ariel Carrasco-Ochoa[2], José Fco. Martínez-Trinidad[2], Andrés Gago-Alonso[1], and José E. Medina-Pagola[1]

[1] Advanced Technologies Application Center
7^a ♯ 21406 e/ 214 and 216, Siboney, Playa, C.P. 12200, Havana, Cuba.
{nacosta,agago,jmedina}@cenatav.co.cu
[2] National Institute for Astrophysics, Optics and Electronic.
Luis Enrique Erro No 1, Sta. Ma. Tonantzintla, 72840, Puebla, México.
{nacosta,ariel,fmartine}@ccc.inaoep.mx

Abstract. Nowadays, there has been a meaningful increase in the use of frequent approximate subgraph (*FAS*) mining for different applications, for example, graph classification. However, the great amount of mined patterns is one of the fundamental drawbacks of FAS mining. This drawback has a negative effect in the computational performance of classifiers, especially in large graph databases where the number of frequent patterns could be very high. In this paper, we propose a research proposal driven to obtain FAS mining algorithms capable to compute a representative subset of patterns. The representative pattern set should be identified into the mining process improving the efficiency in time, in comparison with the time required if this identification is performed in a post-processing stage over all patterns computed by a general FAS mining algorithm.

Key words: Approximate graph mining, representative patterns, graph-based classification.

1 Introduction

In data mining, frequent pattern identification has become a meaningful topic with a wide set of applications in several domains of the science [1]. This topic includes different techniques for pattern extraction, where frequent subgraph mining techniques have been highlighted. Using graphs as basic structure allows identifying patterns with spatial and semantic relationships.
Several algorithms have been developed for finding all frequent subgraphs in a graph database [2–6]. Most of these algorithms use exact matching methods for computing the frequent subgraphs, but there are several practical problems where the need to allow some variations in the data arises. This fact is because there are concrete problems where exact matching could not be applied with positive outcome [7, 8]. This means that it is important tolerating certain level

of distortion, semantic variations, vertices or edges mismatched during the frequent pattern search. Thus, it is required to evaluate the similarity between graphs considering approximate matching. In this way, several algorithms have been developed for frequent approximate subgraph (*FAS*) mining, which use different approximate graph matching techniques allowing the detection of frequent subgraphs with some distortions in the data [7–11]. These FAS mining algorithms have been successfully used for supervised classification, where FASs are used as features for representing objects. This approach has been used in several domains of the science as: analysis of biochemical structures [9, 11], analysis of genetic networks [10], analysis of circuits, links and social networks [8], and image classification [7]. However, in most of these applications, usually a large number of frequent subgraphs is computed [12], therefore, discovering interesting patterns in this set of patterns is still a challenge. Several techniques have been proposed for identifying interesting subgraphs, reducing the dimensionality of the identified pattern set, such as: identifying only maximal, cliques, and closed subgraphs, among others. Using only maximal frequent subgraphs instead of using all the patterns is one of the techniques used to avoid redundancy among the computed patterns and consequently for reducing the dimensionality of this set of patterns. A maximal frequent subgraph is a pattern that is not a subgraph of any other frequent subgraph [13]. It is important to highlight that from the frequent maximal subgraphs it is possible to reconstruct the whole set of frequent subgraphs because all of them are summarized into the maximal patterns. However, from the maximal patterns, the information about the support of non-maximal patterns cannot be retrieved. To face this problem, in several applications closed frequent subgraph are used. A closed frequent subgraph is a pattern that does not have any supergraph with the same frequency [14, 15]. Thus, from the closed frequent subgraphs it is possible to reconstruct the whole set of frequent subgraphs including the information about their support. In real applications such as biochemical compounds, clique frequent subgraphs [9, 16] have been used for reducing the amount of mined patterns. A frequent clique subgraph is a pattern where every two vertices are connected by an edge. Using this kind of patterns, specially when the graph collection contains many clique graphs, the amount of patterns is too high.

This paper is structured as follows: in Section 2, some related works about algorithms for computing representative patterns are described. In Section 3, the research problem is presented. In Section 4, the research proposal is discussed. This proposal includes: the research question, the aims, and the expected contributions of this research. Later, in Section 5 we present some preliminary results. Finally, our conclusions are included in Section 6.

2 Related work

Several researchers have turned their attention to the problem of mining maximal, closed or clique patterns in graph collections [9, 14–17]; however, only a few of these works are based on approximate graph matching:

- *APGM* [9] computes the frequent approximate subgraphs that are cliques in a graph collection. This algorithm uses a depth-first search (DFS) approach for building each clique candidate pattern by extending the edges and using the Canonical Adjacency Matrix code (CAM code) of each candidate, sub-isomorphism tests are applied. Also, in this process, a substitution matrix, containing probabilities of interchange between vertex labels, is used. Although the authors suggested that this idea can be extended to edge labels, this algorithm only deals with variations between vertex labels.
- Z. Zou *et al.* [16] propose an algorithm that computes top-k maximal clique subgraphs in an uncertain graph. In this approach, a combination of both, maximal and clique, are used for taking advantage from both approaches. This algorithm uses an exact approach for computing sub-isomorphism between graphs, but during the candidate generation process, each candidate is identified as a clique evaluating the probability that the candidate has of being a clique across all processed graphs. Each graph, processed by this algorithm, is uncertain because it is built taking into account the existence probability over the original graph.

In this paper, we are focused on the approximate approach for graph mining, which allows some semantic variations in vertex and edge labels keeping the graph topology.

3 Research problem

Frequent approximate subgraph mining have become a very commonly used technique in data knowledge extraction, which has been successfully applied in several domains of the science. This technique has become an important topic in those mining tasks where the mined patterns are detected taking into account distortions in the data. Using these approximate techniques, better results than the exact techniques are reported in some tasks of graph classification, however, it has a main problem that a high number of patterns are identified during the mining process. This high amount of patterns increases the computational resources needed for storing them, affecting the efficiency and efficacy of the methods where they will be used.

4 Proposal

In this section, a research proposal to give a solution to the previously commented problem is presented.

4.1 Research question

Is it possible to propose new algorithms for computing representative frequent approximate subgraphs, that allow keeping or improving the classification efficacy reported in the state-of-the-art when this type of subgraphs are used as attributes in supervised problems?

4.2 Aims

The general aim of this research is:

To propose new algorithms for mining representative FAS that allow us keeping or improving the classification efficacy, in supervised problems, reported in the state-of-the-art when this type of subgraphs are used as attributes.

The specific aims are:

1. Propose a new algorithm for computing maximal FASs in graph collections.
2. Propose a new algorithm for computing closed FASs in graph collections.
3. Propose a new algorithm for computing clique FASs in graph collections.
4. Extend a based-graph classification framework for evaluating the efficacy and efficiency of the representative subgraphs computed by our algorithms.

4.3 Expected contributions

The expected contributions of this proposal are:

1. A review of algorithms for frequent subgraph mining in graph collections.
2. An algorithm for computing the maximal FASs in a graph collection.
3. An algorithm for computing the closed FASs in a graph collection.
4. An algorithm for computing the clique FASs in a graph collection.
5. A graph classification framework based on FAS mining using the proposed algorithms.

5 Preliminary results

As preliminary results of the proposed research, we propose an algorithm for computing maximal frequent approximate subgraphs (*M-FASs*) based on an algorithm for FASM proposed by Acosta-Mendoza *et al.* [7] (VEAM), where substitution matrices are used to specify which vertices, edges or labels can replace some other ones; allowing variation into the vertex and edge labels, but keeping the graph topology.

Our proposal, which is a modification of the VEAM algorithm, called *M-VEAM*, extracts only the maximal FASs from a graph collection. M-VEAM (see Algorithm 1) starts finding the frequent approximate single-edge set C, using a breadth-first search (*BFS*). Later, for each pattern in C, a function "Search" (see Algorithm 2) that recursively computes all extensions of a given pattern using depth-first search (*DFS*), is invoked and if the extended pattern is maximal (i.e. none of its extensions is frequent) then it is stored into the output set F. The function "appLset" (see Algorithm 3) searches the possible approximate label set for the new edge e which is an extension of a pattern T and the possible label set of the new vertex that e connects with an existing vertex in T (if is necessary). Finally, when all FASs in C have been extended, the set F of all M-FASs in the

Algorithm 1: *M-VEAM*

Input: D : A graph collection
MV : Substitution matrix indexed by L_V
ME : Substitution matrix indexed by L_E
τ : Similarity threshold
δ : Support threshold.
Output: F : Maximal frequent approximate subgraph set.

```
1 F ← ∅;
2 C ←the frequent approximate single-edge set in D;
3 foreach T ∈ C do
4     Search(T, D, MV, ME, τ, δ, F);
5     if T is maximal then
6         Insert T in F;
```

Algorithm 2: *Search*

Input: $T = (V_t, E_t, I_t, J_t)$: A frequent approximate subgraph
D : Graph collection
MV : Substitution matrix indexed by L_V
ME : Substitution matrix indexed by L_E
τ : Similarity threshold
δ : Support threshold
F : Frequent approximate subgraph set.
Output: F : Maximal frequent approximate subgraph set.

```
1  foreach o_j ∈ O(T, G_i), where G_i ∈ D do
2      foreach e = ExtSet(o_j) do
3          CL ← appLSet(T, MV, ME, G_i, o_j, e, τ);
4          foreach (elabel, vlabel) ∈ CL do
5              The candidate X is built using the tuple (elabel, vlabel);
6              C ← C ∪ {(X, codeCAM(X), score)};
7  foreach T_1 ∈ C do
8      if sup_G(T_1, D) ≥ δ and T_1 ∉ F then
9          Search(T_1, D, MV, ME, τ, δ, F);
10         if T_1 is maximal then
11             Insert T_1 in F;
```

given collection is returned. More details about VEAM algorithm can be found in [7, 18].

In order to show the usefulness of using M-FASs for image (graph) classification, a comparison between the use, as attributes, of all patterns computed by VEAM [7] against the M-FASs computed by M-VEAM for image classification, is shown. Using the M-FASs computed by M-VEAM we build attribute vectors to represent the images of the collection. An image is represented as an attribute

Algorithm 3: *appLSet*

Input: T : A candidate graph
MV : Substitution matrix indexed by L_V
ME : Substitution matrix indexed by L_E
$G = (V, E, I, J)$: A graph of the collection
G' : Embedding of T in G
$e = \{u, v\}$: An extension of G'
τ : similarity threshold.
Output: CL : A set of candidate 2-tuples $(elabel, vlabel)$.

```
1 foreach j ∈ U_E^τ(J(e)) do
2     scoreE ← S_max(T, G') * ME_{j,J(e)} / ME_{j,j};
3     if e is a forward extension of G' then
4         foreach i ∈ U_V^τ(I(v)) do
5             if i is less than or equal to the largest of the vertex labels of T then
6                 score ← scoreE * MV_{i,I(v)} / MV_{i,i};
7                 if score ≥ τ then CL ← CL ∪ {(j, i)};
8     else if scoreE ≥ τ then CL ← CL ∪ {(j, ∅)};
```

vector $V = (v_1, \ldots, v_n)$ where the number of columns n is the amount of maximal patterns computed by M-VEAM. The value of each attribute v_i $(1 \leq i \leq n)$ is the maximum similarity between the pattern i and the image. Thus, a matrix where the row number is the number of graphs (images) in the collection is built, and the element of each row is the attribute vector which represents the corresponding image.

Two image databases are used in this experiment: *GREC* [19] that contains images of electronic and architectonic plane symbols grouped into 22 classes. This database was split into 572 (52%) images for training and 528 for testing; and *CoenenDB* that contains synthetic images, taken from the Random image generator of Coenen [3], that represents two landscape views; CoenenDB was split into 1200 (60%) images for training and 800 for testing. In both databases, each image is represented as a graph: in GREC, several critical points were selected and used as vertices to build a graph and the edges contain vertex spacial information. For CoenenDB, a tree for each image using a quad-tree method [20] was created and the information of the leaves of these trees was used to build a graph.

In Table 1, the number of patterns used as attributes for classification are compared. These patterns are obtained using $\tau = 40\%$ in the CoenenDB database and $\tau = 8\%$ in the GREC database. These values were computed as the mean of the similarities among the graphs of the collection. This table is split into two sub-tables, one for CoenenDB and for GREC collections, respectively. The first column of each sub-table shows the support value used, and the other two

[3] www.csc.liv.ac.uk/∼frans/KDD/Software/ImageGenerator/imageGenerator.html

consecutive columns show the number of patterns (computed by M-VEAM and VEAM algorithms respectively) used as attributes for classification, and the third column shows the reduction percentage achieved using only maximal patterns.

Table 1. Number of patterns used as attributes in the classification process.

CoenenDB				GREC			
support (δ)	**M-VEAM**	**VEAM**	**Reduction**	**support (δ)**	**M-VEAM**	**VEAM**	**Reduction**
20%	437	745	41.34%	2%	1190	1422	16.32%
25%	186	330	43.64%	3%	607	715	15.10%
30%	86	143	39.86%	4%	366	437	16.25%

As we can see in Table 1, using the subgraphs computed by M-VEAM produces a reduction in the amount of subgraphs used as attributes for classification, compared against the patterns computed by VEAM. In this table, we can see a reduction ranging from 15% to 43%.

The next experiment evaluates the classification results reached using the maximal FASs computed by M-VEAM compared against the results obtained using all FASs as attributes for classification. We summarize the classification results of our experiments in Table 2, which is subdivided in two sub-tables: one shows the accuracy results and the other shows the F-measure results, in the same order. The first and second columns of these sub-tables show the collection name and the support threshold values used in this experiment, respectively. The other four consecutive columns show the classification results (accuracy or F-measure), for the classifier specified in the top of these columns, using only the M-FASs computed by M-VEAM and all FAS computed by VEAM, respectively. Notice that the best results appear boldfaced.

As we can see in Table 2, the results achieved with our proposal are competitive regarding to the results obtained using all patterns computed by VEAM. In the CoenenDB database, the best classification result was obtained by VEAM using the J48graft classifier with an accuracy of 97.25, and using the Regression classifier, M-VEAM obtained the same value. According to the F-measure, M-VEAM obtained the best result using the Regression classifier with an F-measure of 97.26. In this database, using the patterns computed by M-VEAM as attributes, a reduction of 43% was achieved. In the GREC database, the best classification result was obtained using the patterns computed by VEAM jointly with the SVM classifier, obtaining an accuracy of 94.51, while using the patterns computed by M-VEAM we got an accuracy of 93.61 also with the SVM classifier. In this database, the patterns computed by M-VEAM allow a dimensionality reduction of 16% regarding the number of patterns computed by VEAM.

In addition, in Table 3, we present a statistical comparison for all pairwise comparisons between our proposal using M-FASs as attributes and the option of using all patterns computed by VEAM. For this comparison, we use a significant

Table 2. Classification results (%) using several classifiers.

(a) Accuracy results achieved using several δ values

Collection	δ	J48graft		Decision Table		Regression		SVM	
		M-VEAM	VEAM	M-VEAM	VEAM	M-VEAM	VEAM	M-VEAM	VEAM
CoenenDB	20%	96.38	**97.25**	**95.38**	94.38	**97.25**	96.25	**95.50**	95.38
	25%	95.50	**96.75**	**94.00**	80.13	96.25	**96.38**	93.63	**94.38**
	30%	95.50	**96.50**	**95.63**	95.25	96.38	**96.50**	**95.50**	95.13
Average		95.79	**96.83**	**95.00**	89.92	**96.63**	96.38	94.88	**94.96**
GREC	2%	**53.98**	45.45	**57.77**	33.90	**75.57**	73.48	93.61	**94.13**
	3%	77.52	**82.20**	63.64	**65.72**	77.65	**83.14**	93.37	**94.51**
	4%	77.14	**81.63**	59.33	**68.37**	81.11	**82.95**	92.86	**94.13**
Average		69.55	**69.76**	**60.25**	56.00	78.11	**79.86**	93.28	**94.26**

(b) F-measure results achieved using several δ values

Collection	δ	J48graft		Decision Table		Regression		SVM	
		M-VEAM	VEAM	M-VEAM	VEAM	M-VEAM	VEAM	M-VEAM	VEAM
CoenenDB	20%	96.34	**97.23**	**95.43**	94.49	**97.26**	96.21	**95.51**	95.39
	25%	95.47	**96.73**	**93.94**	82.51	96.22	**96.33**	93.57	**94.35**
	30%	95.43	**96.46**	**95.76**	95.33	96.35	**96.50**	**95.47**	95.06
Average		95.75	**96.81**	**95.04**	90.78	**96.61**	96.35	94.85	**94.93**
GREC	2%	**52.00**	38.00	**16.90**	11.76	55.32	**79.17**	91.91	**93.33**
	3%	81.95	**86.96**	27.54	**28.13**	74.70	**78.43**	**93.63**	89.36
	4%	61.32	**78.43**	25.69	**34.29**	74.68	**76.00**	86.11	**86.96**
Average		65.09	**67.80**	23.38	**24.73**	68.23	**77.87**	**90.55**	89.88

statistical test known as Bergmann test [21]. The value for α used on this test was 0.05.

Table 3. Statistical significance results achieved for different classifiers in two image (graph) collections.

	CoenenDB	**GREC**
Test/Classifier	M-VEAM vs. VEAM	M-VEAM vs. VEAM
J48graft	VEAM	–
Decision-Table	M-VEAM	–
Regression	–	–
SVM	–	–

In table 3, the first column of these sub-tables shows the classifiers used in each comparison and columns 2 and 3 show the results for the CoenenDB and GREC image databases, respectively. These columns show the approach that is significant better than the other according to the Bergman test; the symbol "–" indicates that there is not a statistical significant difference between the results of both approaches.

As we can see from Table 3, the use of M-FASs as attributes is a good option since the dimensionality is reduced and we obtain similar classification results than using all the FASs computed by VEAM.

6 Conclusions

Frequent approximate subgraph mining is a widely used technique in Data Mining applications where there is some distortion into the data. However, usually a large number of frequent patterns is computed. Using only representative patterns as attributes instead of using all the patterns is a technique that can be used to reduce the dimensionality of the object descriptions (representation space). Therefore, the aim of our research work is to develop new algorithms for mining representative FAS that allows us improving the classification efficiency and efficacy when this type of subgraphs are used as attributes.

In this paper, we present, as preliminary results of this research work, a modification of a FAS mining algorithm of the state-of-the-art, for computing only maximal FASs in graph collections. The experiments show that using only maximal patterns as attributes, instead of all patterns computed by VEAM, allows obtaining similar classification results, while reducing the dimensionality and removing redundant patterns from the set of patterns used as attributes for image classification.

As future work, we are going to keep developing this research proposal for achieving the specific objectives and general goals.

Acknowledgment. This work was partly supported by the National Council of Science and Technology of Mexico (CONACyT) through the project grants $CB2008$-106443 and $CB2008$-106366; and the scholarship grant 287045.

References

1. Jiang, C., Coenen, F., Zito, M.: A survey of frequent subgraph mining algorithms. Knowledge Engineering Review (2012)
2. Inokuchi, A., Washio, T., Motoda, H.: An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In: Proceedings of the 2000 European Symposium on the Principle of Data Mining and Knowledge Discovery (PKDD'00), France, Lyon (2000) 13–23
3. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: The 3rd IEEE International Conference on Data Mining, FL, Melbourne (2003) 549–552
4. Yan, X., Huan, J.: gSpan: Graph-Based Substructure Pattern Mining. In: International Conference on Data Mining, Japan, Maebashi (2002)
5. Borgelt, C.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proc. IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan, IEEE Press (2002) 51–58
6. Nijssen, S., Kok, J.: A Quickstart in Frequent Structure Mining can make a Difference. In: The 10th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2004) 647–652
7. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.: Frequent approximate subgraphs as features for graph-based image classification. Knowledge-Based Systems **27** (2012) 381–392

8. Holder, L.B., Cook, D.J., Bunke, H.: Fuzzy substructure discovery. In: ML92: Proceedings of the ninth international workshop on Machine learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1992) 218–223
9. Jia, Y., Zhang, J., Huan, J.: An efficient graph-mining method for complicated and noisy data with real-world applications. Knowledge Information Systems **28**(2) (2011) 423–447
10. Song, Y., Chen, S.S.: Item sets based graph mining algorithm and application in genetic regulatory networks. Data Mining, IEEE International Conference on Volume, Issue (2006) 337–340
11. J.L., Zou, Z., Gao, H.: Mining frequent subgraphs over uncertain graph databases under probabilistic semantics. VLDB J. **21**(6) (2012) 753–777
12. Acosta-Mendoza, N., Gago-Alonso, A., Carrasco-Ochoa, J., Martínez-Trinidad, J., Medina-Pagola, J.: Feature Space Reduction for Graph-Based Image Classification. In: Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP'13). Volume Part I, LNCS 8258., Havana, Cuba, Springer-Verlag Berlin Heidelberg (november 2013) 246–253
13. Kimelfeld, B., Kolaitis, P.: The complexity of mining maximal frequent subgraphs. In: Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, ACM (2013) 13–24
14. Yan, X., Han, J.: ClosedGraph: Mining Closed Frequent Graph Patterns. In: Proc. of the 9th ACM SIGKDD of International Conference on Knowledge Discovery and Data Mining (KDD), Washington, DC (2003) 286–295
15. Takigawa, I., Mamitsuka, H.: Efficiently Mining δ-tolerance Closed Frequent Subgraphs. Machine Learning **82**(2) (2011) 95–121
16. Zou, Z., Li, J., Gao, H., Zhang, S.: Finding top-k maximal cliques in an uncertain graph. In: IEEE 26th International Conference on Data Engineering (ICDE 2010). (2010) 649–652
17. Ozaki, T., Etoh, M.: Closed and maximal subgraph mining in internally and extermally weighted graph databases. In: Proceedings of the IEE Workshops of International Conference on Adbanced Information Networking and Applications., IEEE Computer Society. (2011) 626–631
18. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.: On speeding up frequent approximate subgraph mining. In: Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP'12). Volume LNCS 7441., Buenos Aires, Argentina, Springer-Verlag Berlin Heidelberg (2012) 316–323
19. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning, Orlando, USA (2008) 208–297
20. Finkel, R., Bentley, J.: Quad Trees: A Data Structure for Retrieval on Composite Keys. Acta Informatica **4** (1974) 1–9
21. Bergmann, G., Hommel, G.: Improvements of general multiple test procedures for redundant systems of hypotheses. In: P.Bauer, G. Hommel, and E. Sonnemann, editors, Multiple Hypotheses Testing, Springer, Berlin (1988) 100–115