Image Classification Using Frequent Approximate Subgraphs

Niusvel Acosta-Mendoza, Annette Morales-González, Andrés Gago-Alonso,
Edel B. García-Reyes, and José E. Medina-Pagola

Advanced Technologies Application Center (CENATAV), Havana, Cuba
{nacosta,amorales,agago,jmedina,egarcia}@cenatav.co.cu

Abstract. Frequent approximate subgraph (FAS) mining is used in applications where it is important to take into account some tolerance under slight distortions in the data. Following this consideration, some FAS miners have been developed and applied in several domains of science. However, there are few works related to the application of these types of graph miners in classification tasks. In this paper, we propose a new framework for image classification, which uses FAS patterns as features. We also propose to compute automatically the substitution matrices needed in the process, instead of using expert knowledge. Our approach is tested in two real image collections showing that it obtains good results, comparable to other non-miner solutions reported, and that FAS mining is better than the exact approach for this task.

Keywords: Approximate graph mining, frequent approximate subgraphs, graph-based image representation, image classification.

1 Introduction

Data in multiple domains can be naturally modeled as graphs [11] since graphs are general and powerful data structures that can be used to represent diverse types of objects. Several authors have developed graph-based techniques and methods for satisfying the need of converting large volumes of data into useful information [6]. The frequent approximate subgraph (FAS) discovery is an example of such techniques [1,4,5]. These techniques have become important topics in mining tasks where the mined patterns are detected taking into account distortions in the data.

The aforementioned techniques have been successfully used in several domains of the science. An important area of intelligent data analysis is the development of classifiers using FAS as features. However, from the reported FAS miners, only *APGM* [5] and *VEAM* [1] use FASs as features in classification tasks. APGM is used in both synthetic data set and real data set of protein structure pattern identification and structure classification, while VEAM is used in several synthetic data sets for image classification.

In this work we propose a framework using FAS mining methods for image classification. APGM and VEAM are used to detect the FASs in a graph collection, where the approximation consists in considering some variations of the

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 292–299, 2012.

data through the substitution probability, preserving the topology of graphs. In this paper, we propose a graph-based image representation and a classification framework using FASs as features on a real image collection. We also propose to compute automatically the substitution matrices employed by FAS miners, in contrast to other approaches [1,5] where such task is usually left in hands of human experts. Several graph miners are used to evaluate our proposed framework.

Very few approaches have been reported relating image classification with graph mining techniques. Among them, we find [7,15], where they use frequent subgraph mining to build a vocabulary, following the bag-of words approach [10]. The differences with our proposal lies in the graph-based image representation and in the use of an exact subgraph mining algorithm. We advocate the idea that using FAS is a better choice to model slight image variations.

The basic outline of this paper is as follows. Section 2 provides some basic concepts and the approximate pattern definitions used. The graph-based image representation is presented in Section 3. The framework for image classification is explained in Section 4 and the experimental results in a real image collection are discussed in Section 5. Finally, conclusions of the research and some ideas about future directions are exposed in Section 6.

2 Background

In this section, we start by providing the background knowledge and notation used in the following sections. Next, the definition of approximate patterns, which is the subject of this paper, is showed. Finally, the frequent approximate subgraph mining problem is formalized.

2.1 Basic Concepts

This work is focused on simple undirected labeled graphs; henceforth, when we refer to graph we assume this type of graph. Before presenting their formal definition, we will define the domain of labels.

Let L_V and L_E be label sets, where L_V is a set of vertex labels and L_E is a set of edge labels. The domain of all possible labels is denoted by $L = L_V \cup L_E$.

A *labeled graph* in L is a 4-tuple, $G = (V, E, I, J)$, where V is a set whose elements are called *vertices*, $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is a set whose elements are called *edges* (the edge $\{u, v\}$ connecting the vertex u with the vertex v), $I : V \to L_V$ is a *labeling function* for assigning labels to vertices and $J : E \to L_E$ is a *labeling function* for assigning labels to edges.

Let $G_1 = (V_1, E_1, I_1, J_1)$ and $G_2 = (V_2, E_2, I_2, J_2)$ be two graphs, we say that G_1 is a *subgraph* of G_2 if $V_1 \subseteq V_2$, $E_1 \subseteq E_2$, $\forall u \in V_1, I_1(u) = I_2(u)$, and $\forall e \in E_1, J_1(e) = J_2(e)$. In this case, we use the notation $G_1 \subseteq G_2$.

Given G_1 and G_2, we say that f is an *isomorphism* between these graphs if $f : V_1 \to V_2$ is a bijective function, where $\forall u \in V_1, f(u) \in V_2 \wedge I_1(u) = I_2(f(u))$ and $\forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2 \wedge J_1(\{u, v\}) = J_2(\{f(u), f(v)\})$. When there is an isomorphism between G_1 and G_2, we say that G_1 and G_2 are *isomorphic*.

Let Ω be the set of all possible labeled graphs in L, the *similarity* between two elements $G_1, G_2 \in \Omega$ is defined as a function $sim : \Omega \times \Omega \rightarrow [0,1]$. We say that the elements are very different if $sim(G_1, G_2) = 0$, the higher the value of $sim(G_1, G_2)$ the more similar the elements are and if $sim(G_1, G_2) = 1$ then there is an isomorphism between these elements.

Let $D = \{G_1, \ldots, G_{|D|}\}$ be a graph collection and let G be a labeled graph in L, the *support* value of G in D is obtained through the following equation:

$$supp(G, D) = \sum_{G_i \in D} sim(G, G_i)/|D| \tag{1}$$

If $supp(G, D) \geq \delta$, then the graph G is approximately frequent in the collection D, saying that G is a *FAS* in D. Notice that when we refer to a graph collection we assume that it is the representation built from a real graph collection. The value of the support threshold δ is in $[0,1]$ assuming that the similarity is normalized to 1. *FAS mining* consists in finding all the FASs in a collection of graphs D, using a similarity function sim and a support threshold δ.

2.2 Approximate FAS Methods Considered

In APGM [5] and VEAM [1] algorithms, the idea that not always a vertex label or an edge label can be replaced by any other is upheld. Therefore, these algorithms specify which vertices, edges or labels can replace others using substitution matrices to perform the frequent subgraph mining. APGM only deals with the variations among vertex labels, while VEAM performs the mining process using the vertex and edge label sets. These methods use the substitution matrix that can have a probabilistic interpretation and they offer frameworks for each frequent subgraph mining task.

A *substitution matrix* $M = (m_{i,j})$ is an $|L| \times |L|$ matrix indexed by a label set L. An entry $m_{i,j}$ $(0 \leq m_{i,j} \leq 1, \sum_j m_{i,j} = 1)$ in M is the probability that the label i is replaced by the label j. When M is diagonally dominant (i.e. $M_{i,i} > M_{i,j}, \forall j \neq i$) then M is known as *stable matrix*.

Let $G_1 = (V_1, E_1, I_1, J_1)$ and $G_2 = (V_2, E_2, I_2, J_2)$ be two labeled graphs in L, MV be a substitution matrix indexed by L_V, ME be a substitution matrix indexed by L_E, and τ be the isomorphism threshold. We say that G_1 is *approximate isomorphic* to G_2, denoted by $G_1 =_A G_2$, if there exists a bijection $f : V_1 \rightarrow V_2$ such that:

- $\forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2$,
- $S_f(G_1, G_2) = \prod_{u \in V_1} \frac{MV_{I_1(u), I_2(f(u))}}{MV_{I_1(u), I_1(u)}} * \prod_{e=\{u,v\} \in E_1} \frac{ME_{J_1(e), J_2(\{f(u), f(v)\})}}{ME_{J_1(e), J_1(e)}} \geq \tau$.

The bijection f is an approximate isomorphism between G_1 and G_2, and $S_f(G_1, G_2)$ is the product of normalized probabilities called *approximate isomorphism score* of f. When G_1 is approximate isomorphic to a subgraph of G_2, we say that G_1 is *approximate sub-isomorphic* to G_2. Notice that this is a generalization of the APGM approach [5].

The *approximate matching score* between two graphs, denoted by $S_{max}(G_1, G_2)$, is the largest approximate isomorphism score.

$$S_{max}(G_1, G_2) = max_f\{S_f(G_1, G_2)\} \tag{2}$$

Given a graph collection D and an isomorphism threshold τ, the *approximate support* of a graph G, denoted by $supp(G, D)$, is the average score of the graph in the collection, where G is approximate isomorphic to a subgraph of graphs in the collection:

$$supp(G, D) = \sum_{G_i \in D} S_{max}(G, G_i)/|D| \tag{3}$$

If $supp(G, D) \geq \delta$, then the graph G is approximately frequent in the collection D, saying that G is a *frequent approximate subgraph* in D, with δ as support threshold. Notice that the values of the products of normalized probabilities $S_f(G_1, G_2)$ is in the interval $(0, 1]$. The value of the support threshold δ is in $[0, 1]$ assuming that $S_{max}(G, G_i)$ is normalized. The *frequent subgraph mining* task used in this paper consists in finding all the connected frequent subgraphs in a collection of graphs D, using (3), δ as support threshold, and τ as isomorphism threshold.

3 Graph-Based Image Representation

In order to use graph mining techniques for image classification, it is necessary to obtain a graph-based image representation. For this purpose, we used the approach presented in [11,12]. We construct an irregular pyramid for each image [2], which provides a hierarchy of partitions at different levels of resolution. Each level is a region adjacency graph (RAG) where each region of the partition is a vertex in this graph, and an edge exists between two vertices, if the underlying regions are adjacent. The pyramid is built from bottom to top, being the base level (level 0) the whole image (i.e. each vertex of the base level represents one pixel in the image, and the edges are the 4-connections of each pixel). Each level l is constructed from its previous level $l - 1$, by means of contraction kernels, which are sets of vertices in level $l - 1$ that are selected to be contracted into a surviving vertex. In the new level l, each surviving vertex will represent all the vertices from level $l - 1$ in its contraction kernel, and will keep a connection to them. Further information regarding the construction of the pyramid can be obtained in [2,8].

Once we have the pyramid for an image, its vertices and edges are labeled using the image regions and graph information at each level. The vertices, which represent regions, will contain a color histogram which will be computed using 16 bins per channels in the RGB color model, yielding a 48 bin histogram. Also, local binary patterns (LBP) [14] will represent the texture information of the region, distributed into a 256 bin histogram. The edges will store the spatial descriptor (binary vector) proposed in [11], representing several topological and orientation relationships between pairs of regions. This graph labeling corresponds to the one used in [11,12].

Each image is represented by a single graph, therefore, in order to select which level of he pyramid should be selected, we used the B measure proposed in [12]. This measure evaluates each level of the pyramid against a border map of the image, in terms of how much each partition preserves the borders present in the map. The best level evaluated by B is selected to represent the image.

3.1 Automatically Building Substitution Matrices

In order to compute the substitution matrix for the vertices, it is necessary to reduced the set of vertices labels. According to the pyramid representation explained before, there will be as much labels as possible pairs of different color and texture histograms. To reduce the set of vertices labels, we use a clustering algorithm to group similar features. The centroid of each cluster will be the new label of all the vertices with features belonging to this cluster. Then, the substitution matrix will be a $n \times n$ matrix, where n is the number of labels (clusters). Each element of this matrix will store the similarity between two labels, given by the similarity between the centroids of the clusters they belong to. In this case, we decided to use the Euclidean distance between the concatenation of the color and LBP histogram for each node. This means that an element of this matrix can be interpreted as the confidence to substitute a node with label x with a node with label y in a matching scheme.

The substitution matrix for edges is easier to construct, since using the spatial descriptor representation we can have only 27 possible configurations of spatial relations. The value that will be stored in the elements of the matrix is obtained by the Sokal-Michener measure proposed in [11] for computing the similarity between spatial descriptors.

4 Classification Framework

Given a set of pre-labeled real images, we obtain the graph collection that represents these images by producing the graph-based image representations presented in Section 3. After that, the FAS miners are used to obtain all the FASs of the mentioned graph collection. The FASs extracted from the graph collection are considered an analogy to the vocabulary obtained in the bag-of-features approach, converting our proposal in a sort of bag-of-subgraphs approach. Having this vocabulary composed of subgraphs, the feature vectors of the original images are built using those FASs as features. The dimension of the new feature will be the number of FASs found in the collection. In our framework, as well as proposed by [1], the feature vectors are built taking into account the approximation values in each image of the collection. That means that for every subgraph in the vocabulary, if it is present in an image, then its corresponding value in the new feature is the highest similarity value of its occurrence in the image.

When all the new features are built, a classifier generator (SVM using 10 cross-validation) is used having such vectors as data to produce an image classifier. The complete flowchart of our classification framework is shown in Fig. 1.

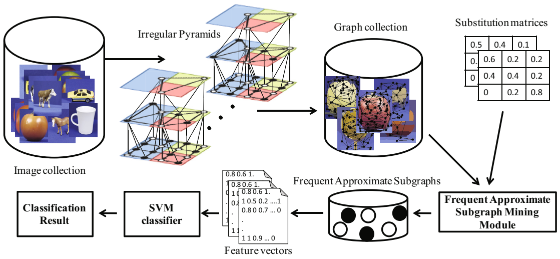

Fig. 1. Framework of graph-based image classification

5 Experimental Results

We chose two well known databases to test our approach: the COIL-100 [13] and the ETH-80 [9] image sets. Both databases contain images of simple objects taken from different viewpoints. We represented all images by a single graph, which corresponds to the "best" segmented level of each pyramid (See Section 3). The COIL-100 image set is a database of color images of 100 objects having 72 poses per object. In Figure 2 some examples are shown. We took 25 objects randomly selected from this dataset to test our classification framework. The ETH-80 Image Set database contains 80 objects from 8 categories. Each object is represented by 41 different views yielding a total of 3280 images (See Figure 2). This database is more challenging than the COIL-100 database in the sense of the viewpoint diversity. For the experiment in this database we took the same 6 categories employed by [11]: *apples*, *cars*, *cows*, *cups*, *horses* and *tomatoes*.

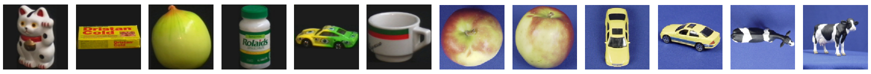

Fig. 2. Example images from the COIL-100 Image Set database (first 6 images), and from the ETH-80 Image Set database (last 6 images)

The results of the experiment are presented in Table 1. In this table we can see the comparison of our framework using three different graph miners, i.e, gdFil [3], APGM and VEAM. The first one represents the exact methods and the last two are FAS miners. Columns show different support thresholds (δ) used in the experiments. For the case of APGM and VEAM, the isomorphism threshold was set to $\tau = 0.4$. The first thing to notice is that the approximate graph miners achieve a higher accuracy in most cases than the exact ones, showing the relevance of allowing slight differences in real data. Regarding the approximate methods, for the case of the COIL-100 database, we can see that the VEAM obtained better accuracies than APGM, which indicates that the use of the edge distortion in the FAS mining can provide additional cues for classification. In the ETH-80 dataset, the edge distortions did not provided any relevant information though.

Table 1. Accuracies achieved by gdFil, APGM and VEAM algorithms

a) COIL database using 25 random classes.

	Support (δ)						
Algorithm	**70%**	**60%**	**50%**	**40%**	**30%**	**20%**	**10%**
gdFil	-	-	-	-	23.06%	60.39%	**85.89%**
APGM	**21.94%**	**57.11%**	90.28%	91.39%	90.94%	87.83%	84.33%
VEAM	**21.94%**	**57.11%**	**91.35%**	**92.18%**	**91.52%**	**89.24%**	84.69%

b) ETH-80 database using 6 classes.

	Support (δ)						
Algorithm	**70%**	**60%**	**50%**	**40%**	**30%**	**20%**	**10%**
gdFil	-	-	-	-	28.70%	47.80%	76.63%
APGM	**26.75%**	**31.67%**	**51.91%**	**82.03%**	**82.03%**	**82.03%**	**81.38%**
VEAM	**26.75%**	**31.67%**	**51.91%**	**82.03%**	**82.03%**	81.83%	76.54%

We compared our proposal with other classification methods that do not use FAS mining techniques. In the COIL-100 dataset, the method proposed by [12] obtained 91.6% while our method scored 92.18%. For the case of ETH-80 dataset, our method obtained 82.03%, which is comparable to other state-of-the-art methods according to the comparison performed by [12], where the results range from 76% to 88%.

These results show that the proposed framework, which involves using FAS mining and automatically computing the substitution matrices (and not using expert knowledge in this process), is able to provide good outcomes for real image classification.

6 Conclusions

In this paper we proposed a framework for image classification using FASs as features, which are obtained using FAS miners reported in the literature. They are able to detect FAS patterns in graph collections allowing slight semantic differences among graphs. Within our framework, we also propose to use substitution matrices computed automatically based on image features, which proves that not using expert knowledge for this task can also produce good results. The graph-based image representation was extracted from irregular graph pyramids, relabeling the vertices using clustering techniques. Since our approach is an application of FAS mining for real graph-based collections, the classification accuracy results obtained by traditional miners are smaller than the obtained by FAS miners in most cases. Also, the experimental results show that our proposal is comparable with other state-of-the-art methods for image classification.

As future work, we are going to develop new ways for taking advantage of FAS selection strategies for improving graph classification (such as, using discriminative FASs, representative FASs, etc.). These strategies in combination with FAS miners could be useful for reducing dimensionality and improving the efficiency of graph classifiers.

References

1. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.E.: Frequent Approximate Subgraphs as Features for Graph-Based Image Classification. Knowledge-Based Systems 27, 381–392 (2012)
2. Brun, L., Kropatsch, W.: Introduction to combinatorial pyramids. In: Digital and Image Geometry: Advanced Lectures, pp. 108–128 (2001)
3. Gago-Alonso, A., Carrasco-Ochoa, J.A., Medina-Pagola, J.E., Martínez-Trinidad, J.F.: Full Duplicate Candidate Pruning for Frequent Connected Subgraph Mining. Integrated Computer-Aided Engineering 17, 211–225 (2010)
4. Holder, L.B., Cook, D.J., Bunke, H.: Fuzzy Substructure Discovery. In: Proceedings of the Ninth International Workshop on Machine Learning, San Francisco, CA, USA, pp. 218–223 (1992)
5. 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), 423–447 (2011)
6. Jiang, C., Coenen, F., Zito, M.: A Survey of Frequent Subgraph Mining Algorithm. To appear in Knowledge Engineering Review (2012)
7. Jiang, C., Coenen, F.: Graph-based image classification by weighting scheme. In: Proceedings of the Artificial Intelligence, pp. 63–76. Springer, Heidelberg (2008)
8. Kropatsch, W., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. Pattern Recognition Letters 26(3), 319–337 (2005)
9. Leibe, B., Schiele, B.: Analyzing Appearance and Contour Based Methods for Object Categorization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003), pp. 409–415 (2003)
10. Li, F.F., Perona, P.: A Bayesian Hierarchical Model for Learning Natural Scene Categories. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), pp. 524–531 (2005)
11. Morales-González, A., García-Reyes, E.: Assessing the Role of Spatial Relations for the Object Recognition Task. In: Bloch, I., Cesar Jr., R.M. (eds.) CIARP 2010. LNCS, vol. 6419, pp. 549–556. Springer, Heidelberg (2010)
12. Morales-González, A., García-Reyes, E.B.: Simple object recognition based on spatial relations and visual features represented using irregular pyramids. In: Multimedia Tools and Applications, pp. 1–23. Springer, Netherlands (2011), `http://dx.doi.org/10.1007/s11042-011-0938-3`
13. Nene, S., Nayar, S., Murase, H.: Columbia Object Image Library (COIL-100). Technical Report, Department of Computer Science, Columbia University CUCS-006-96 (1996)
14. Ojala, T., Pietikainen, M.: A comparative study of texture measures with classification based on featured distribution. Pattern Recognition 29(1), 51–59 (1996)
15. Ozdemir, B., Aksoy, S.: Image Classification Using Subgraph Histogram Representation. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, pp. 1112–1115 (2010)