

Feature Space Reduction for Graph-Based Image Classification

Niusvel Acosta-Mendoza^{1,2,*}, Andrés Gago-Alonso¹,
Jesús Ariel Carrasco-Ochoa², José Fco. Martínez-Trinidad²,
and José E. Medina-Pagola¹

¹ Advanced Technologies Application Center (CENATAV),
7a # 21406 e/ 214 and 216, Siboney, Playa, CP: 12200, Havana, Cuba
{nacosta,agago,jmedina}@cenatav.co.cu

² National Institute of Astrophysics, Optics and Electronics (INAOE),
Luis Enrique Erro No. 1, Sta. María Tonantzintla, Puebla, CP: 72840, Mexico
{ariel,fmartine}@inaoep.mx

Abstract. Feature selection is an essential preprocessing step for classifiers with high dimensional training sets. In pattern recognition, feature selection improves the performance of classification by reducing the feature space but preserving the classification capabilities of the original feature space. Image classification using frequent approximate subgraph mining (FASM) is an example where the benefits of features selections are needed. This is due using frequent approximate subgraphs (FAS) leads to high dimensional representations. In this paper, we explore the use of feature selection algorithms in order to reduce the representation of an image collection represented through FASs. In our results we report a dimensionality reduction of over 50% of the original features and we get similar classification results than those reported by using all the features.

Keywords: Approximate graph mining, approximate graph matching, feature selection, graph-based classification.

1 Introduction

Finding a discriminative subset of features is essential when there are high dimensional representations. Feature selection algorithms allow improving classifiers performance by reducing the feature space and keeping discrimination capabilities of the original representation. The main idea of these algorithms is to calculate a subset of the input features by removing those with little or no predictive information for classification [3–6, 10, 15, 17, 19]. These algorithms can be arranged into three main groups: wrapper algorithms [3], filter algorithms [6, 10, 19] and embedded algorithms [5, 17]. Wrapper ones use a classifier to evaluate feature subsets. The advantage of these algorithms is the interaction between the feature subset search and the classifier, but it is an expensive

* Corresponding author.

process. Filter ones evaluate the feature subsets without involving any classifier during selection and, generally, they are faster than wrapper algorithms. Embedded ones combine the advantage of wrappers and filters including some interaction with the classifier.

FASM has become a technique of great significance in mining tasks where the frequent subgraphs are computed considering data distortions. The usefulness of the patterns computed by these algorithms has been shown in different classification tasks [1, 2, 12, 14]; but only a few of them have been applied to image classification [1, 2] outperforming the results of exact algorithms [7, 18]. However, using a large number of graphs (patterns) as features could affect the performance of the classifiers, due to the high dimensionality of the representation of the images.

In this paper, we explore the use of feature selection to reduce the representation of an image collection represented through FASs. Experiments show that our proposal allows drastically reducing the dimensionality of feature vector while getting competitive results compared with the method that uses all the features for classification. On the other hand, to the best of our knowledge, this is the first work that uses a fusion of FASM and feature selection algorithms for image classification.

The organization of this paper is the following. In Section 2, some basic concepts are presented. In Section 3, we describe how the feature selection is included into the framework for image classification using FASM. Later, in Section 4, some experiments are shown in order to empirically validate the efficiency and accuracy of feature selection for feature space reduction in graph-based image classification. Finally, our conclusions and future directions are discussed in Section 5.

2 Background

This work is focused on collections of simple undirected labeled graphs. Henceforth, when we refer to a graph we are assuming this type of graphs.

A *labeled graph* in the domain of all possible labels $L = L_V \cup L_E$, where L_V and L_E are the label sets for vertices and edges respectively, 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 $\{u, v\}$ connects the vertex u with the vertex v), $I : V \rightarrow L_V$ is a *labeling function* for assigning labels to vertices and $J : E \rightarrow L_E$ 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 $G_1 \subseteq G_2$ and we also say that G_2 is a *supergraph* of G_1 .

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

If there is an isomorphism between G_1 and G_2 , we say that G_1 and G_2 are *isomorphic*. If G_1 is isomorphic to G_3 and $G_3 \subseteq G_2$, then we say that there is a *sub-isomorphism* between G_1 and G_2 , and we also say that G_1 is sub-isomorphic to G_2 .

Let $D = \{G_1, \dots, G_{|D|}\}$ be a collection of graphs and G be a labeled graph in L , the *support* value of G in D is defined as the fraction of graphs $G_i \in D$, such that there is a sub-isomorphism between G and G_i . This value of support is obtained using the following equation: $supp(G, D) = |\{G_i \in D: G \text{ is sub-isomorphic to } G_i\}|/|D|$

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

As there are several correspondences between two graphs, $sim_{max}(G_1, G_2) = max\{sim(G_1, G_2)\}$ is defined as the highest value of similarity which can be obtained between the different correspondences of G_1 and G_2 .

Let $D = \{G_1, \dots, G_{|D|}\}$ be a graph collection and G be a labeled graph in L , the approximate support (denoted by $appSupp$) value of G in D , in terms of the similarity, is computed as: $appSupp(G, D) = \sum_{G_i \in D} sim_{max}(G, G_i)/|D|$.

When $appSupp(G, D) \geq \delta$, then G is a *frequent approximate subgraph* (FAS) in D . The value of the support threshold δ is in $[0, 1]$ since the similarity is defined in $[0, 1]$. *Frequent approximate subgraph mining* consists in finding all the FAS in a collection of graphs D , using a similarity function sim and a support threshold δ .

3 Proposed Framework

Given a pre-labeled image collection, we built a graph collection for representing these images, following the same approach proposed in [1]. The FAS are obtained from the graphs that represent the images in the collection, using a FASM algorithm. These patterns (FAS) are used as features for representing each image of the collection. In fact, feature vectors for the original images are built using such patterns, in the same way as in [1], taking into account the similarity of each image of the collection (represented as a graph) to each pattern. Later, these feature vectors are reduced by using a feature selection algorithm (information gain, chi-squared, and gain ratio feature evaluation) in order to take into account only the selected features for classification.

Following the main idea of [1], the proposed framework for graph-based image classification including feature selection consist of four modules (see Figure 1): representation module, pattern extraction module, feature selection module, and classification module. Representation, pattern extraction and classification modules are the same ones respectively used in [1]. The feature selection module is the main contribution of our proposal. This module is introduced in order to identify better features for representing the classes of the image collection than those used in [1].

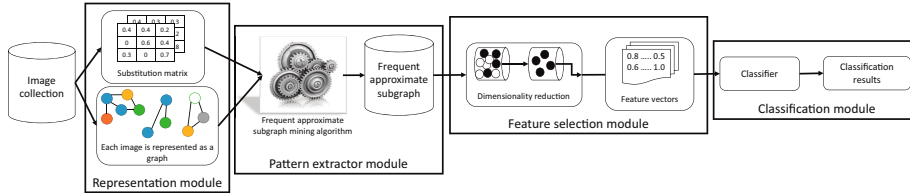


Fig. 1. The graph-based image classification framework proposed in this paper

4 Experiments

In this section, some experiments to show the impact of feature selection in graph-based image classification using FASM are presented. These experiments include a comparison between the framework for graph-based image classification including feature selection and the framework proposed in [1] which uses all the FASs for representing the images. The FASM algorithm, which we select to be used in the pattern extraction module for our experiments, is VEAM [1]. This decision was made based on the results reported in [1], where the use of patterns obtained taking into account distortion in vertex and edge labels, i.e. those computed by VEAM, are better for image classification than: those patterns found by APMG [14].

4.1 Databases

In order to compare the results obtained by our proposal against those reported in [1], the image collection used in this paper is the same. This collection consists of 700 images obtained by the Random image generator of Coenen¹. These images are divided in two classes “landscape” and “seascape”, according to their content. The process to represent these images as graphs is the same used in [1], which is based on quad-trees with 4 as depth limit for divisions. This collection was split into six sub-collections with different sizes: from 200 to 700 images with an increment of 100 images. All these collections contain 18 vertex labels, 24 edge labels and a mean graph size ranging between 43 to 47 in terms of the number of edges.

4.2 Experimental Results

The comparison presented in this section is performed over the graph collections detailed in Section 4.1. Once we have the FAS’s computed by VEAM, a feature selection algorithm is used only once to obtain the feature subset that represents each class in the collection. Finally, this subset of features is used as input for a classifier using 10-fold cross-validation.

¹ www.csc.liv.ac.uk/~frans/KDD/Software/ImageGenerator/imageGenerator.html

In Table 1, six cells where each represents a collection database are shown. Each of these cells show a comparison between the number of features (patterns) used in the classification process. Four columns are grouped into each the mentioned cells, where the first of these columns shows the patterns computed by VEAM, the other three columns show the number of patterns that will be selected by each filter feature selector per each classifier algorithm. The number of features selected were obtained experimentally in a range [50,200], which resulted as the best options after carrying out several experiments. Finally, each classifier is specified in the last columns.

In our experiments, we use several classifiers to evaluate our proposal. These classifiers are of different nature: Support Vector Machine (SVM); Bayesian network (BayesNet); decision trees (J48graft); and boosting (AdaBoost). All these classifiers, except SVM, were taken from Weka v3.6.6 [9] using the default parameters. For SVM, we used the same libSVM used in [1]. Moreover, we use three filter feature selection algorithms: information gain (IG), chi-squared (CHI-Q), and gain ratio feature evaluation (GRAE). In these experiments, we compared the accuracy reached by selecting a feature subset computed by VEAM through a filter feature selector against the accuracy obtained by using all the features computed by VEAM as it is proposed in [1].

Table 1. Number of features used in the classification process

Coenen-200 ($\delta = 20\%$)				Coenen-300 ($\delta = 20\%$)				Coenen-400 ($\delta = 20\%$)				Classifier	
All	IG	CHI-Q	GRAE	All	IG	CHI-Q	GRAE	All	IG	CHI-Q	GRAE		
340	200	133	200	374	140	110	200	433	200	185	185	SVM	
	110	80	95		125	110	125		125	80	65	65	BayesNet
	155	125	155		140	50	140		140	150	150	150	AdaBoost
	133	110	110		125	140	155		125	110	140	140	J48graft
Coenen-500 ($\delta = 25\%$)				Coenen-600 ($\delta = 20\%$)				Coenen-700 ($\delta = 20\%$)				Classifier	
All	IG	CHI-Q	GRAE	All	IG	CHI-Q	GRAE	All	IG	CHI-Q	GRAE		
238	140	155	155	498	185	95	200	864	200	200	200	SVM	
	65	50	50		155	65	50		65	80	65	65	BayesNet
	155	101	110		95	80	125		65	65	155	155	AdaBoost
	95	101	95		65	200	125		200	200	200	200	J48graft

The results of our experiments are summarized in Table 2. This table is divided in two sub-tables, one for a pair of filter feature selectors. The first and second columns of these sub-tables show the name of the collection and the support threshold value that got the best results for this collection. The other eight columns are grouped in four, each group associated to a different supervised classifier, the first column in each group shows the accuracy obtained by using all the features (patterns) computed by VEAM, while the other three columns show the accuracy obtained by using a features selected by the filter feature selector specified in the top of the column.

As we can see in Table 2, the results achieved using our framework using feature selection are competitive in accuracy with those results using the framework proposed in [1], and it is important to highlight that our proposal uses less than 50% of features used by the framework proposed in [1] in most cases.

In addition, in Table 3 we present an statistical comparison of the tested classifiers using all the features (patterns) computed by VEAM against using the features selected by each one of the filter feature selectors. For this comparison, we use two significant statistical tests [8]: Holm [13], and Hommel [11] tests. The value for α used on these tests is 0.05.

Table 2. Accuracy results achieved using different classifiers in several graph (image) collections with and without the use of several feature selection algorithms.

Collection	δ	SVM (linear kernel)				BayesNet			
		All	IG	CHI-Q	GRAE	All	IG	CHI-Q	GRAE
Coenen-700	20%	95.86%	96.29%	96.43%	96.43%	90.29%	94.57%	94.57%	94.57%
Coenen-600	20%	95.83%	96.50%	96.17%	96.50%	91.17%	94.83%	94.67%	94.50%
Coenen-500	25%	97.20%	97.40%	97.60%	97.60%	90.60%	94.80%	94.80%	94.80%
Coenen-400	20%	96.75%	96.50%	96.75%	97.25%	93.25%	95.25%	95.50%	95.50%
Coenen-300	20%	97.33%	97.00%	97.00%	97.00%	88.33%	95.00%	95.00%	95.00%
Coenen-200	20%	97.50%	97.00%	95.50%	97.50%	88.00%	94.50%	94.50%	94.50%
Average		96.75%	96.78%	96.58%	97.05%	90.27%	94.83%	94.84%	94.81%

Collection	δ	AdaBoost				J48graft			
		All	IG	CHI-Q	GRAE	All	IG	CHI-Q	GRAE
Coenen-700	20%	94.14%	94.29%	94.43%	94.14%	96.14%	96.29%	96.43%	96.43%
Coenen-600	20%	92.67%	94.33%	94.17%	93.67%	95.67%	94.50%	96.17%	96.00%
Coenen-500	25%	94.80%	94.80%	94.80%	94.80%	95.80%	96.60%	96.40%	96.60%
Coenen-400	20%	94.50%	94.75%	94.75%	95.25%	94.50%	96.00%	95.75%	95.50%
Coenen-300	20%	95.00%	95.00%	95.33%	95.00%	94.33%	96.00%	94.67%	95.33%
Coenen-200	20%	94.00%	94.50%	95.00%	94.50%	91.50%	95.00%	95.00%	95.00%
Average		94.19%	94.61%	94.75%	94.56%	94.66%	95.73%	95.74%	95.81%

Table 3. Significant statistical tests (Holm and Hommel) using $\alpha = 0.05$ for different classifiers in several graph (image) collections using all the features computed by VEAM and using the features selected by the filter feature selectors.

Classifier	SVM	BayesNet	AdaBoost	J48graft
All vs. GRAE	-	GRAE	-	GRAE
All vs. CHI-Q	-	CHI-Q	CHI-Q	CHI-Q
All vs. IG	-	IG	IG	IG
IG vs. GRAE	-	-	-	-
CHI-Q vs. GRAE	-	-	-	-
IG vs. CHI-Q	-	-	-	-

In the first column of Table 3, “All” represents the approach using all features computed by VEAM while IG, CHI-Q and GRAE represent our approach using the feature selection through information gain, chi-squared, and gain ratio, respectively. The other columns show what approach is significantly better than the other one; the symbol “-” indicates that there is not a statistical significant difference between the results of both approaches.

As we can see from Tables 3 and 2, the use of the feature selection algorithm CHI-Q is the best option since in 75% of the results it is significantly better than “All”, and using CHI-Q, the better classification accuracies are obtained in general. The other feature selection algorithms IG and GRAE outperform “All” in 75% and 50% of the results, respectively. And in general, IG, CHI-Q and

GRAE are significantly similar of the tests. Between CHI-Q and IG, we select CHI-Q as the best option because it had a better performance in term of the accuracy results.

5 Conclusions

In this paper, we propose the use of approximate subgraphs jointly with feature selection for image classification. To our best knowledge, this is the first work that performs such combination. Using a feature subset for representing the images in the collection allows us to drastically reduce the dimensionality of the feature vectors. This reduction was more than 50% in most cases. Moreover, our proposal is competitive in accuracy regarding the method that uses all the FASs.

As future work, we are going to study the identification of representative features (patterns) during the FASM process. Thus, only the discriminative patterns will be computed during the mining step, improving the effectiveness of FAS classifiers and reducing the runtime for the training stage.

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. 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. Acosta-Mendoza, N., Morales-González, A., Gago-Alonso, A., García-Reyes, E.B., Medina-Pagola, J.E.: Image Classification Using Frequent Approximate Subgraphs. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) *CIARP 2012*. LNCS, vol. 7441, pp. 292–299. Springer, Heidelberg (2012)
3. Bermejo, P., de la Ossa, L., Gámez, J.A., Miguel-Puerta, J.: Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. *Knowledge-Based Systems* 25(1), 35–44 (2012)
4. Bolón-Canedo, V., Sánchez-Marño, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. *Knowledge and Information Systems* 34(3), 483–519 (2013)
5. Duval, B., Hao, J.K., Hernandez, J.C.: A memetic algorithm for gene selection and molecular classification of cancer. In: *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pp. 201–208. ACM, Montreal (2009)
6. Ferreira, A.J., Figueiredo, M.A.T.: Efficient feature selection filters for high-dimensional data. *Pattern Recognition Letters* 33(13), 1794–1804 (2012)
7. Gago-Alonso, A., Carrasco-Ochoa, J.A., Medina-Pagola, J.E., Martínez-Trinidad, J.F.: Duplicate Candidate Elimination and Fast Support Calculation for Frequent Subgraph Mining. In: Corchado, E., Yin, H. (eds.) *IDEAL 2009*. LNCS, vol. 5788, pp. 292–299. Springer, Heidelberg (2009)

8. García, S., Herrera, F.: An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1) (2009)
10. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. *Advances in Neural Information Processing Systems* 18, 507–514 (2006)
11. Hommel, G.: A stagewise rejective multiple test procedure. *Biometrika* 75, 383–386 (1988)
12. Holder, L., Cook, D., Bunke, H.: Fuzzy substructure discovery. In: *Proceedings of the 9th International Workshop on Machine Learning, San Francisco, CA, USA*, pp. 218–223 (1992)
13. Holm, S.: A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6, 65–70 (1979)
14. 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)
15. Norshafarina, O.B., Fantimatufaridah, J.B., Mohd-Shahizan, O.B., Roliana, I.B.: Review of feature selection for solving classification problems. *Journal of Research and Innovation in Information Systems*, 54–60 (2013)
16. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. *Pattern Recognition Letters* 15, 1119–1125 (1994)
17. Rodríguez-Bermúdez, G., García-Laencina, P.J., Roca-González, J., Roca-Dorda, J.: Efficient feature selection and linear discrimination of (eeg) signals. *Neurocomputing* 115(4), 161–165 (2013)
18. Yan, X., Huan, J.: gSpan: Graph-Based Substructure Pattern Mining. In: *Proceedings International Conference on Data Mining, Maebashi, Japan*, pp. 721–724 (2002)
19. Ye, Y., Wu, Q., Huang, J.Z., Ng, M.K., Li, X.: Stratified sampling for feature subspace selection in random forests for high dimensional data. *Pattern Recognition* 46(3), 769–787 (2013)