Graph Clustering Through Attribute Statistics Based Embedding

Jaume Gibert[1], Ernest Valveny[2], Horst Bunke[3] and Luc Brun[1]

[1] École Nationale Supérieure d'Ingénieurs de Caen, ENSICAEN
Université de Caen Basse-Normandie, 6 Boulevard Maréchal Juin
14050 Caen, France
{jaume.gibert,luc.brun}@ensicaen.fr
[2] Computer Vision Center, Universitat Autònoma de Barcelona
Edifici O Campus UAB, 08193 Bellaterra, Spain
ernest@cvc.uab.es
[3] Institute for Computer Science and Applied Mathematics, University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
bunke@iam.unibe.ch

Abstract. This work tackles the problem of graph clustering by an explicit embedding of graphs into vector spaces. We use an embedding methodology based on occurrence and co-occurrence statistics of representative elements of the node attributes. This embedding methodology has already been used for graph classification problems. In the current paper we investigate its applicability to the problem of clustering color-attributed graphs. The ICPR 2010 Graph Embedding Contest serves us as an evaluation framework. Explicit and implicit embedding methods are evaluated in terms of their ability to cluster object images represented as attributed graphs. We compare the occurrence and co-occurrence based embedding methodology to explicit and implicit embedding techniques proposed by the contest participants and show that improvements are possible. We then demonstrate further improvements by means of different vectorial metrics and kernel functions on the embedded graphs.

1 Introduction

Clustering, or unsupervised learning, is a key concept in pattern recognition. While the clustering of vectorial pattern representations has reached some level of maturity, the clustering of graphs is still in its infancy [5]. This is due to a number of difficulties that arise especially from the fact that many operations needed in a clustering algorithm, although readily available for vectorial representations, do not exist for graphs (or are at least extremely difficult to accomplish). Examples are the computation of the mean of a set of graphs, or the operation of making two graphs more similar to each other, as needed in kMeans clustering and self-organizing nets, respectively. In order to overcome these problems, a number of approaches have been proposed that relate the graph domain to vector spaces where such operations are easier to perform and plenty of learning machinery

is available. Graph embeddings and graph kernels are the main paradigms. The former explicitly assign a feature vector to each graph while the latter implicitly map each graph in a feature space and compute the corresponding scalar product. The relation between graph embeddings and graph kernels is clear since, given an explicit embedding, any kernel function on vectors also defines a graph kernel.

We have previously proposed an explicit embedding approach which is based on extracting features describing the occurrence and co-occurrence of node label representatives in a given graph [4]. Its efficiency and good performance, when compared to state of the art methodologies for graph classification, have been empirically demonstrated. In the current paper, we aim at an evaluation of this embedding methodology for graph clustering. To that end, we make use of the ICPR 2010 Graph Embedding Contest [3]. This contest was organized in order to provide a framework for direct comparison between embedding methodologies for the purpose of graph clustering. Three object image datasets were chosen and converted into graphs, divided into a training and a test set. The participants also received a code with which they could assess their own methodologies in terms of a clustering measure. Object images were first segmented into different regions and a region adjacency graph was constructed. Each node representing a region was attributed with the corresponding relative size and the average RGB color components, while edges remained unattributed.

For the contest, four algorithms were submitted, three explicit embedding methods and an implicit one. Jouili and Tabbone assign a feature vector to every graph by considering the eigenvectors of a positive semidefinite matrix regarding the dissimilarity of graphs [6]. Riesen and Bunke map every graph to a feature vector whose components are the edit distances to a predefined set of prototypes [8]. Luqman et at. encode relevant information by quantizing node and edge attributes via the use of fuzzy intervals [7]. Finally, the implicit methodology proposed by Osmanlıoğlu *et al.* maps each node of each graph to a vector space by means of the caterpillar decomposition, and computes a kernel value between two given graphs in terms of a point set matching algorithm based on the Earth Mover's distance [2].

The contribution of the work described in this paper is to evaluate the novel embedding methodology of [4], which was not yet available at the time of the ICPR contest, for the task of clustering and compare it to existing approaches. Besides, the mentioned embedding methodology has been formulated in such a way that it can handle color-based attributed graphs. We will show that, in such a way, it constitutes an attractive addition to the set of graph clustering tools currently available. For the purpose of self-completeness, Section 2 of the paper provides a brief introduction to graph embedding using node label occurrence and co-occurrence statistics. Next, Section 3 describes in detail the experimental evaluation and shows how to gain further improvements along this line of research. Finally, Section 4 draws conclusions from this work.

2 Attribute Statistics based Embedding

The main idea of the embedding methodology used in this work is based on counting the frequency of appearance of the node labels in a given graph and also on the co-occurrence of pairs of node labels in conjunction with edge linkings. The fact that node labels might not be discrete (as it is in the present case) demands for a discretization of the node labelling space and, thus, the selection of a set of representatives. Under the proposed approach, the features are obtained by computing statistics on these representatives in terms of those nodes that have been assigned to each of them. Based on how this assignment from nodes to representatives is made we have two formulations of the embedding approach.

2.1 Hard Assignment

Assume a set of graphs $\mathcal{G} = \{g_1, \ldots, g_N\}$ is given, each being a four-tuple $g_i = (V_i, E_i, \mu_i, \nu_i)$ consisting on a set of nodes V_i, a set of edges $E_i \subseteq V_i \times V_i$, and the corresponding labelling functions μ_i and ν_i. In this work, nodes are labelled with RGB values, thus the labelling function codomain is always the set $[0, 255]^3$ (the relative size attribute is disregarded). Edges remain unlabelled.

From the set of all node labels of all graphs in $\mathcal{G}$ we select some representatives $\mathcal{W} = \{w_1, \ldots, w_n\}$ (see Section 2.3). Given a graph g, each node $v \in V$ is assigned to the closest representative by

$$\lambda_h(v) = \underset{w_i \in \mathcal{W}}{\operatorname{argmin}} \parallel \mu(v) - w_i \parallel_2 . \tag{1}$$

Then we extract unary features as occurrences of representatives in the graph by

$$U_i = \#(w_i, g) = |\{v \in V \,|\, w_i = \lambda_h(v)\}|. \tag{2}$$

Also, co-occurrence features between two representatives are defined as

$$\begin{aligned} B_{ij} &= \#(w_i \leftrightarrow w_j, g) \\ &= |\{(u, v) \in E \,|\, w_i = \lambda_h(u) \wedge w_j = \lambda_h(v)\}|. \end{aligned} \tag{3}$$

Both the unary features U_i and the binary ones B_i are eventually arranged in a feature vector.

In particular, note that what this formulation is proposing is to build a histogram of the presence of specific features in the graphs. In the present case, we aim at evaluating the presence of each color in every graph, and also the presence of the neighbouring relations of all colors in the graphs. In Section 2.4 we discuss the connections of this approach to other existing graph embedding methodologies.

2.2 Soft Assignment

Assigning nodes to representatives in a hard fashion might lead to weak descriptions because the graph extraction process is usually noisy. However, the

embedding methodology used in this work is adaptable for a fuzzy assignment from nodes to representatives which might correct such situations. In particular, each node is represented by a set of probabilities

$$\lambda_s(v) = (p_1(v), \ldots, p_n(v)), \tag{4}$$

where $p_i(v) = P(v \sim w_i)$ is the probability of node v being represented by w_i. The unary features are then the addition of all probabilities for all nodes in the graph g:

$$U_i = \#(w_i, g) = \sum_{v \in V} p_i(v). \tag{5}$$

The fuzzy version of the binary features needs to regard the transition probabilities from one node to the other and thus is defined as

$$\begin{aligned} B_{ij} &= \#(w_i \leftrightarrow w_j, g) \\ &= \sum_{(u,v) \in E} p_i(u)p_j(v) + p_j(u)p_i(v). \end{aligned} \tag{6}$$

2.3 Representative Set Selection

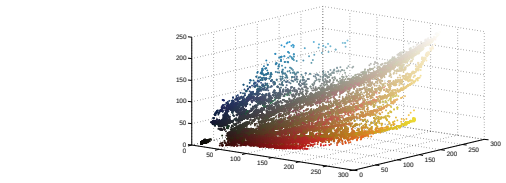
One of the key issues of the embedding methodology is the selection of the set of representatives for the node labels. We can make use of generic clustering approaches independent to the domain such as kMeans, or we can use domain-specific approaches. In this work we use, in addition to kMeans, a color-based approach that tries to adapt the set of representatives to the inherent RGB structure of the node labelling space.

Generic Approaches In order to select representatives of the node labels for the hard assignment version of the proposed embedding, we use the kMeans clustering algorithm for different values of the parameter k. This representation will be referred to as *Hard kM*. In Fig. 1(a) we show a sample of node labels with their corresponding original color. Next to it, in Fig. 1(b), the distribution of the $k = 10$ clusters after applying the kMeans algorithm is shown.

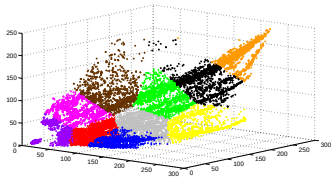
For the soft assignment version we use fuzzy kMeans to select the set of representatives, and the probability of each node to belong to a certain representative is defined to be inversely proportional to the Euclidean distance between the considered node and the representative. We will refer to this representations as *Soft kM*.

These two configurations not only depend on how the representative elements are selected, but also on the number of them. This parameter needs to be validated using the training set.

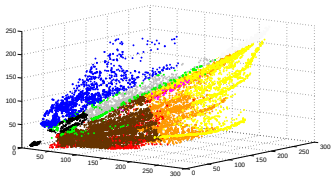
Color-based Approaches Due to the spherical arrangement of its clusters, kMeans does not really account for the color distribution of the original node



(a) Original Color



(b) kMeans Clusters



(c) Color Naming

Fig. 1. Distributions of the graphs' nodes in the RGB space (best seen in color). (a) Original color of each node. (b) kMeans clusters for $k = 10$. (c) Color naming distribution.

values, grouping for instance node labels of different color into the same cluster. This problem demands for a way of selecting representatives that can adapt in a more accurate manner to the real RGB distribution. To do so, we have adopted a color naming approach for which each node label, *i.e.* each point in the RGB space, is assigned to one of the eleven basic colors of the color naming theory.

In particular, we adopted the methodology proposed in [1], where each RGB point is automatically assigned to every color in the color naming scheme with a certain probability. Each node is thus represented with a set of probabilities allowing the use of the soft assignment version of the embedding approach described here. This version is referred to in the text as *Soft Color*. For the hard assignment version, for every node in a given graph, we just pick the color that produces the highest probability value and refer to it as *Hard Color*. Fig. 1(c) shows the resulting 11 clusters after the assignment.

2.4 Relation to other approaches

An interesting observation to be done regarding the proposed methodology is its connections to other graph characterization approaches. In particular, an appealing consideration is that of its similarity with fingerprint characterization of molecular structures [9]. In this domain, molecules are represented as histograms of the presence of particular subgraph structures, where such substructures are selected based on prior chemical knowledge. The methodology used in this work is related to this one in the sense that, after node attribute discretization, it look for particular substruces in the graph representations such as node appearences and node-edge-node walks.

Table 1. C-index on the test sets under the Euclidean distance: lower index values indicate better clustering results. Comparison with the contest participants. The best results are shown bold face.

Embedding	ALOI	COIL	ODBK	Geometric Mean
Osmanlıoğlu et al.	0.088	**0.067**	0.105	0.085
Jouili and Tabbone	0.136	0.199	0.138	0.155
Riesen and Bunke	**0.048**	0.128	0.132	0.093
Luqman et al.	0.379	0.377	0.355	0.370
Hard kM	0.080	0.160	0.070	0.096
Soft kM	0.068	0.136	0.058	0.081
Hard Color	0.067	0.143	0.061	0.083
Soft Color	0.056	0.121	**0.051**	**0.070**

On the other hand, but strictly related to the former, the explicit embedding of graphs by the presented approach can be reduced to characterize graphs based on walks of length 0 and walks of length 1 with respect to their labelling information. In that sense, it might also be connected to the so-called family of random walk kernels [10].

3 Experimental Evaluation

The three object image datasets that were used in the contest are the ALOI, COIL and OBDK collections. Each of them is representing object images under different angles of rotation and illumination changes. For more details on the datasets, we refer to the contest report [3]. We recall here that a training and a test set are available for each dataset. We use the training set to validate the parameters (number of representative elements) that are eventually used for processing the test set.

Every approach is assessed by computing the C-index clustering measure, and approaches are ranked in terms of the geometric mean of the results on the three datasets. When the embedding is explicit, the clustering index is computed based on the Euclidean distances of the vectorial representations of graphs. When an implicit formulation is given, distances are computed according to the following formula

$$d_{ij} = \sqrt{k_{ii} + k_{jj} - 2k_{ij}} \tag{7}$$

where k_{ij} is the kernel value between graphs g_i and g_j. Under a kernel function, graphs are implicitly mapped to a hidden feature space where the scalar product is calculated. Formula (7) is the Euclidean distance between the corresponding vectors in such a feature space. Results of the described embedding methodologies in comparison with the contest participants are shown in Table 1.

As expected, the *Soft* approaches obtain better results than the hard ones, and the color-based versions improve the generic ones. Compared to the participants methods, the proposed embedding approach ranks second on two databases and first on the third one. This leads to the best geometric mean among all tested methods. Moreover, let us mention the high efficiency of our approach

Table 2. C-index under different distances and under the k_{χ^2} kernel on the test sets. The best results are shown bold face.

Distance / Kernel	ALOI		COIL	
	Soft kM	*Soft Color*	*Soft kM*	*Soft Color*
L_2	0.073	0.056	0.136	0.121
L_1	0.064	0.060	0.130	0.110
χ^2	**0.031**	0.032	0.066	0.064
k_{χ^2}	0.088	0.083	**3.10e-08**	9.04e-07
	ODBK		Geometric Mean	
	Soft kM	*Soft Color*	*Soft kM*	*Soft Color*
L_2	0.056	0.051	0.083	0.070
L_1	0.063	0.061	0.081	0.074
χ^2	0.033	0.037	0.041	0.042
k_{χ^2}	**8.67e-10**	0.097	**1.33e-06**	1.94e-03

which arises from the fact that we base our embedding method on very simple features with a fast computation.

In other works, the proposed embedding methodology has been shown to perform better under different vectorial metrics than the Euclidean distance [4]. We refine our results by computing the C-index for clustering validation under the L_1 and χ^2 distances. Results of these experiments for the *Soft* versions are shown on the first three rows of Table 2 (*Hard* versions are discarded since they do not show as good a performance as the *Soft* ones). The χ^2 distance is providing the best results, ranking best on all datasets, even when compared to the contest participants (we, however, want to point out that a direct comparison to the results obtained by the contest participants would not be fair since we do not know how their algorithms would perform under other metrics). Interestingly, the χ^2 distance extracts the best out of the *Soft kM* versions since it outperforms the *Soft Color* one in two of the three datasets, which does not happen when using the two other metrics.

Finally, in order to relate our methodology to those that provide an implicit embedding of graphs we compute kernel values between embedded graphs as

$$k_d(g_1, g_2) = \exp\left(-\frac{1}{\gamma} d(\phi(g_1), \phi(g_2))\right), \quad \gamma > 0 \tag{8}$$

where $\phi(g_i)$ is the vectorial representation of the graph g_i under the described embedding methodology, and d is the χ^2 metric (L_2 or L_1 could also be used but χ^2 is the one providing the best results when clustering under metrics as discussed above). Distance values for the C-index computation are calculated using Eq. (7). The γ parameter is also validated using the training set. Results for the *Soft* versions are shown on the last row of Table 2.

Although the results for the ALOI database worsen when using the kernel values for both versions of the embedding, the most significant point to highlight from this table is that we obtain almost perfect separation indexes for the COIL

dataset under the two *Soft* versions and also for the ODBK under the *Soft kM* one. This makes the geometric means to drastically decrease and demonstrates the embedding methodology we propose in this work being a strong approach for graph clustering.

4 Conclusions

In this work, we have evaluated the explicit graph embedding methodology that accounts for statistics on node label representatives in terms of clustering performance. We have compared it to the approaches that were reported in the ICPR 2010 Graph Embedding Contest and shown that it performs very favorably. Additional improvements are gained by evaluating the embedding method under different metrics and also by the use of kernel functions on the resulting vectors, leading to almost perfect separation results in two of the three contest datasets. As a final remark, the authors find of paramount importance that different works appear in the same line as the present, where different pattern recognition methodologies are brought together and compared ones to the others using a unified and clear framework. In this sense, we want to acknowledge the ICPR Graph Embedding Contest organizers for their work.

References

1. R. Benavente, M. Vanrell, R. Baldrich, *Parametric fuzzy sets for automatic color naming* J. Optical Society of America A, 25 (10), pp. 2582-2593 (2008).
2. M.F. Demirci, Y. Osmanlıoğlu, A. Shokoufandeh, S. Dickinson, *Efficient many-to-many feature matching under the l_1 norm*, Computer Vision and Image Understanding 115 (7) (2011), pp. 976-983.
3. P. Foggia, M. Vento, *Graph Embedding for Pattern Recognition*. ICPR 2010, LNCS 6388, Springer, 2010, pp. 75-82.
4. J. Gibert, E. Valveny, H. Bunke, *Graph embedding in vector spaces by node attribute statistics*. Pattern Recognition 45 (9) (2012), pp. 3072-3083.
5. A.K. Jain, *Data Clustering: 50 years beyond K-means*. Pattern Recognition Letters 31 (8) (2010), pp. 651-666.
6. S. Jouili, S. Tabbone, *Graph Embedding Using Constant Shift Embedding*, ICPR 2010, LNCS 6388, Springer, 2010, pp. 83-92.
7. M.M. Luqman, J. Lladós, J.-Y. Ramel, T. Brouard, *A Fuzzy-Interval Based Approach for Explicit Graph Embedding*, ICPR 2010, LNCS 6388, Springer, 2010, pp. 93-98.
8. K. Riesen, H. Bunke, *Graph Classification and Clustering Based on Vector Space Embedding*. World Scientific (2010).
9. P. Mahé, N. Ueda, T. Akutsu,J.-L. Perret, J.-P. Vert, *Graph Kernels for Molecular Structure-Activity Relationship Analysis with Support Vector Machines*. Journal of Chemical Information and Modelling (2005), pp. 939-951.
10. T. Gärtner, P. Flach, S. Wrobel, *On graph kernels: hardness results and efficient alternatives* Proc. 16th Annual Conference on Learning Theory, (2003), pp. 129-143.