Median graph shift: A new clustering algorithm for graph domain

Salim Jouili, Salvatore Tabbone
LORIA-INRIA UMR 7503,
BP 239, 54506 Vandoeuvre-lès-Nancy Cedex, France
{*salim.jouili, tabbone*}*@loria.fr*

Vinciane Lacroix
Royal Military Academy, Signal and Image Centre
Avenue de la Renaissance 30, B-1000 Bruxelles, Belgium
Vinciane.Lacroix@elec.rma.ac.be

***Abstract*—In the context of unsupervised clustering, a new algorithm for the domain of graphs is introduced. In this paper, the key idea is to adapt the mean-shift clustering and its variants proposed for the domain of feature vectors to graph clustering. These algorithms have been applied successfully in image analysis and computer vision domains. The proposed algorithm works in an iterative manner by shifting each graph towards the median graph in a neighborhood. Both the set median graph and the generalized median graph are tested for the shifting procedure. In the experiment part, a set of cluster validation indices are used to evaluate our clustering algorithm and a comparison with the well-known Kmeans algorithm is provided.**

I. Introduction

Graphs give a universal and flexible framework to describe the structure and the relationship between objects. They are useful in many different application domains like pattern recognition, computer vision and image analysis. Generally, classical document retrieval systems produce a ranked list of documents in response to the query document. In the case of graph-based representation, the query is a graph which represents the query document. If this query is general, it is difficult to identify the specific graphs which the user is interested in. Consequently, a natural alternative to ranking is to cluster the target set into groups of graphs with common aspects. Clustering aims to synthesize a huge amount of data by a small number of homogeneous and distinct clusters, such that all objects in the same cluster are similar to each other and the objects the most dissimilar belong to different clusters. A lot of clustering algorithms have been proposed in the literature, a major part of these algorithms deal with data represented by feature vectors. We refer the reader to the Xu's survey [19]. Whereas, just a few works are interested to structural-based data representation, in particular graphs [4]. These works fall roughly into two categories. The first category contains the methods for which a mapping from the domain of graphs to feature vectors are proposed. Almost all these methods use the notion of the graph kernel [13] to embed a graph into a feature vector, for example, by means of dissimilarities to some prototype graphs. Then, classical clustering techniques are applied to graphs embedded into vectors. The second category involves directly the work in the graph domain, indeed the proposed approaches include the computation of the representatives of clusters [18] or the use of median graph notion [9] to adapt classical clustering techniques into the domain of graphs [10].

In this paper we propose a new graph clustering algorithm by making use of a seeking mode in the same philosophical vein as the mean [2], median [12], [11] and medoid [17] shift clustering techniques in the domain of feature vectors. However, in the domain of graphs, the computation of mean or median of a set of graphs can not be performed with the same easiness as in domain of vectors. In fact, computing a distance between two graphs is in itself an open problem. This problem is usually referred to as the graph edit distance which is considered to be a NP-Complete problem and requires an exponential time and space to find optimal solution [1]. Nevertheless, to cope this problem many approaches have been proposed to approximate the graph edit distance, we refer the interested reader to the survey in [3]. Based on these approximation techniques, some new notions that compute medians and representatives of a set of graphs have been proposed. For all these issues the median graph [9] has grown on as the efficient candidate to represent the center of a set of graphs. In this work, this notion is used to implement the shifting operation instead of the mean used in the classical mean-shift clustering. In other words, the proposed algorithm works in an iterative manner by shifting each graph towards the median graph of graphs in its neighborhood. Like mean-shift, the median graph shift computes the number of clusters during execution. In the experiment part, a set of cluster validation indices are used to evaluate our clustering algorithm. In addition, a comparison with the well-known Kmeans algorithm is provided.

II. Preliminaries

A. Median shift clustering

Mean-shift clustering [2] is a popular mode seeking algorithm that offers a non-parametric approach which does not require *a priori* knowledge of the cluster's number, and does not set any restrictions on the shape of the clusters. An interesting variants of the mean-shift algorithm is the median shift [12], [11] in which the data points are shifted towards the median instead of the mean as follows. Let X=$\{x_1, \cdots, x_N\}$ be a set of points embedded in a n-dimensional Euclidean space, and $S_i \subseteq X$ be the set of data point x_j in n-sphere characterized by its radius *h* and centered on x_i. Then, $\forall x_j \in S_i \; \|x_i - x_j\| < h$. At each iteration all data point in X are considered in parallel: the data points $x_j \in S_i$ are considered for median computation, shifting x_i (the center of S_i) towards the median point m_i which will be considered for next iteration. Let us note that the convergence has been proved in [11]. In addition, it has been shown empirically in [12] that the median-shift procedure converges faster than the mean-shift.

B. Graph edit distance and median graph

Matching by minimizing edit distance gauge the distance between graphs by counting the least cost of edit operations needed to make two graph isomorphic. A standard set of edit operations is given by insertions, deletions and substitutions. These edit operations are applied on both edges and nodes. In addition, a certain cost is associated with each of these operations. Obviously, for every pair of graphs A and B there exists different sequence of edit operations transforming A into B. However, the computation of the edit distance between two graphs involves not only finding a sequence of edit operations to transform one graph to the other, but also finding such a sequence that possesses the minimum total cost.

Definition Let A=(V$_a$,E$_a$) and B=(V$_b$,E$_b$) be two graphs. The graph edit distance between A and B is given by:

$$d(A,B) = \min_{(e_1,..,e_k)\in\gamma(A,B)} \sum_{i=1}^{k} c(e_i)$$

where $\gamma(A,B)$ denotes the sequences of edit operations transforming A into B and $c(e_i)$ denotes the cost of the edit operation e$_i$.

In order to compute a optimal graph edit distance, several techniques have been proposed. In a recent work, Riesen and al. [16] propose an approximate computation of the graph edit distance by means of bipartite graph matching, In the experiment part of this paper we use this approach.

The median graph is introduced by Jiang and Bunke in [9] and refined in recent works (e.g. [6]). It is a useful tool to compute a representative of a set of graphs. We distinguish two kinds of median graph, the set median graph (SM) $\hat{g}$ and the generalized median graph (GM) $\bar{g}$. The set median graph is a graph belonging to the involved set of graphs S. Whereas, the generalized median graph is a graph belonging to the set of graphs U that can be constructed using the labels in the initial set S. Formally speaking,

$$\hat{g} = \min_{g\in S} \sum_{g_i\in S} d(g,g_i), \text{ and } \bar{g} = \min_{g\in U} \sum_{g_i\in S} d(g,g_i)$$

where d is a distance between graphs. Obviously, it is difficult to compute the generalized median graph, as its time complexity grows exponentially with the size of U. In this paper we use the approximate generalized median graph via embedding [6]. The set median graph, can be computed by a straightforward procedure with a complexity O(n^2).

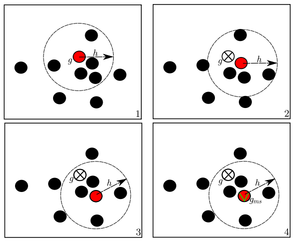

Figure 1. Execution of the repeat-until loop

III. MEDIAN GRAPH SHIFT

A pseudo-code description of the proposed algorithm is given in Algorithm 1. The algorithm can serve two goals, either clustering or selecting representative prototypes. In this paper, we draw an evaluation of its clustering application. From a given set of graphs G, the algorithm returns a set of clusters $\{C_j\}$ and each cluster has a representative prototype p_i. The radius *h*, called bandwidth in the classical mean-shift, is a parameter fixed *a priori*. The algorithm computes the number of cluster during execution. In the algorithm, first each graph $g_i \in G$ is associated to an empty graph g_{msi} (line 1). Then, for each graph $g_i \in G$ the inner loop (line 3-7) is performed. This loop computes for the graph g_i a steady median graph g_{msi}. We define the steady median graph g_{msi} as the final median graph returned by a shifting series of g_i. In the experiments, it has been shown that this process converges. To compute a steady median graph for a graph g_i, only a subset $G_i \subseteq G$ centered on g_i with a radius h is considered (line 4). Then g_i is shifted (line 6) towards the median graph of G_i which is computed by an external procedure $median()$ (line 5). Figure 1 illustrates an execution of the repeat-until loop, in this example the convergence of the graph g to the steady median graph g_{ms} is done in four iterations. That is, after each iteration through the repeat loop, the subset G_i is more compact than the previous iteration because of the substitution of its center by a median graph which minimizes the sum of distance in G_i, and consequently keep the convergence of the algorithm. Hence, the final steady median graph g_{msi} can be regarded as a cluster convergence graph. The cluster around g_{msi} consists of exactly those graphs that converge by shifting to g_{msi}. Finally, the result is generated (line 10) as follows: the number of clusters is the number of distinct g_{msi}. Note here that, we consider two graph g_1 and g_2 as distinct if their distance $d(g_1,g_2) \neq 0$. Next, the set of prototypes P is defined from the distinct steady median graphs g_{msi}. Each cluster C_j is composed by the graphs $g_z \in G$ that converge to p_j.

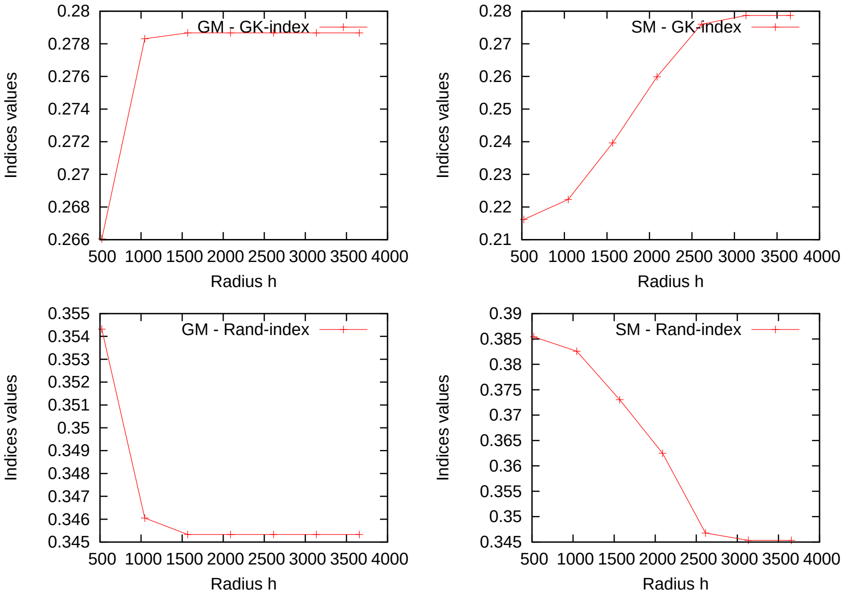

Figure 2. Different validation indices in function of the radius h, for SM (left), GM (right) with GREC Data set

Algorithm 1 Median Graph Shift pseudo-code

```
Require: A set of graphs, G=$\{g_1, \cdots, g_n\}$, and a radius $h$
Ensure: A set of clusters $\{C_j\}_{j=1}^{k}$, and a set of prototypes P=$\{p_1, \cdots, p_k\}$
Associate to each $g_i \in$ G an empty graph $g_{ms_i}$
for each $g_i \in$ G do
    repeat
        let $G_i \subseteq$ G, where $\forall g_k \in G_i$, d($g_i$,$g_k$)<$h$
        $g_m \leftarrow median(G_i)$        ▷ Median graph computation
        Shift $g_i$ towards $g_m$
    until $g_i$ converge to a steady median graph ($g_m$ does not change)
    $g_{ms_i} \leftarrow g_m$
end for
Assign graphs with the same steady median graph to the same cluster C$_j$, where 1 <j< $k$ and k is number of distinct $g_{ms_i}$.
```

IV. Experimental results

To perform the evaluation of the proposed algorithm, we used the Mutagenicity (Molecules), the Letter (distorted letter drawings) and the GREC (symbols from architectural and electronic drawings) datasets from [15]. The experiments consisted in applying our algorithm for each dataset using the set median graph and the generalized median graph [6] which are two possible implementations of the procedure $median()$ in Algorithm 1. In addition, in order to evaluate the impact of the radius h on the results, we performed several repetitions of each experiment with different values of h. This value varies from the minimum distance between two graphs to the maximum in each data set. Then, the clustering performance was evaluated using two cluster validation indices, the $Goodman$-$Kruskal$ $index$ [7] and the $Rand$ $index$ [14]. These indices have been previously used in the context of graph clustering in [8], [5]. Figure 2 shows the results of the clustering indices on the GREC dataset by changing the value of the radius h. In left column: two curves of the values of different validation indices as function of the value of radius h, where the used median procedure is the generalized median graph [6], and the set median graph in the right column. For the remainder, we assume that the $best$ radius h consists of the value which maximize the two indices since high values of each index value correspond to a better clustering. Concretely, for each value of h we sum the two indices and we take the h value's that maximizes this sum.

Table I summarizes the best radius to each data set with their corresponding indices values using the set median graph and the generalized median graph and provides a comparison with the Kmeans algorithm. Here, the graph edit distance approximation and the set median graph are used in the Kmeans algorithm to compute the centers and to perform the clustering. Let us recall that the Kmeans algorithm is not deterministic. That is, the clustering result achieved by Kmeans depend on the k initial selected graphs which are selected randomly. That is why, we performed 10 repetitions on each data set and we take the average value of each cluster validation index. We observe that the proposed algorithm outperforms the Kmeans clustering on all the data set regarding the GK-index. Regarding $Rand$

		GREC	Mutagenicity	Letter
Best h	SM	1567.1	14.389	3.059
	GM	1044.7	9.592	0.764
Indices				
GK	SM	0.239	0.306	0.189
	GM	**0.278**	**0.621**	**0.578**
	Kmeans	0.234	0.164	-0.236
$Rand$	SM	**0.373**	**0.539**	0.544
	GM	0.346	0.5	0.232
	Kmeans	0.363	0.512	**0.924**

Table I
Comparison with the results of Kmeans algorithm. (Best selected radius h, SM: set median, GM: generalized median)

index, the median graph shift algorithm achieves the best result for two data sets.

In addition to the non-parametric and deterministic properties of the median graph shift algorithm, the clustering results are better than the Kmeans algorithm regarding separability and compactness. Regarding similarity to the ground truth the proposed algorithm outperforms Kmeans for two data sets.

V. Conclusion

In this paper, we consider the clustering of graphs. A new graph clustering algorithm is proposed. It is an adaptation of the well-established mean-shift algorithm into domain of graphs. The notion of set median and generalized median graph is used to implement the shifting operation instead of the mean in the classical mean-shift clustering. The median graph shift clustering is a deterministic and non-parametric algorithm. It computes the number of clusters during execution. We have performed a set of clustering experiments with three data sets using two validation indices. The results have shown that the proposed clustering algorithm is able to produce meaningful clustering for graphs set. In a near future, we will focus our works to discuss deeply the issue of the bandwidth selection by adapting previous works developed for the mean-shift algorithm.

Acknowledgment

This research is partially supported by the French National Research Agency project NAVIDOMASS referenced under ANR-06-MCDA-012 and Lorraine region.

References

[1] H. Bunke. Recent developments in graph matching. In *ICPR*, pages 2117–2124, 2000.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 24:603–619, 2002.

[3] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[4] R. Englert and R. Glantz. Towards the clustering of graphs. *IAPR Workshop GbRPR 1999, Austrian Computer Society*, pages 125–133, 1999.

[5] M. Ferrer, E. Valveny, F. Serratosa, I. Bardají, and H. Bunke. Graph-based *K*-means clustering: A comparison of the set median versus the generalized median graph. In *CAIP, LNCS 5702*, pages 342–350, 2009.

[6] M. Ferrer, E. Valveny, F. Serratosa, K. Riesen, and H. Bunke. An approximate algorithm for median graph computation using graph embedding. In *19th Intl. Conf. on ICPR.*, pages 1–4, Dec. 2008.

[7] L. Goodman and W. Kruskal. Measures of association for cross-classifications. *J. American Statistical Association*, 1954.

[8] S. Gunter and H. Bunke. Validation indices for graph clustering. *IAPR Workshop GbRPR 2001*, pages 229–238, 2001.

[9] X. Jiang, A. Mnger, and H. Bunke. On median graphs: Properties, algorithms, and applications. *IEEE TPAMI*, 23(10):1144–1151, 2001.

[10] S. Jouili and S. Tabbone. A hypergraph-based model for graph clustering: Application to image indexing. In *13th Intl. Computer Analysis of Images and Patterns, LNCS 5702*, pages 360–368, 2009.

[11] V. Lacroix. Automatic palette identification of colored graphics. In *IAPR Workshop on Graphics Recognition*, pages 95–100, 2009.

[12] V. Lacroix. Raster-to-vector conversion: Problems and tools towards a solution a map segmentation application. In *7th Intl. Conf. on Advances in Pattern Recognition, ICAPR*, pages 318–321, Feb. 2009.

[13] B. Luo, R. C. Wilson, and E. R. Hancock. Spectral feature vectors for graph clustering. *IAPR Workshop on S+SSPR, LNCS 2396*, pages 83–93, 2002.

[14] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[15] K. Riesen and H. Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *IAPR Workshop on S+SSPR, LNCS 5342*, pages 287–297, 2008.

[16] K. Riesen and H. Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image Vision Comput.*, 27(7):950–959, 2009.

[17] Y. Sheikh, E. A. Khan, and T. Kanade. Mode-seeking by medoidshifts. In *IEEE 11th Intl. Conf. on Computer Vision, ICCV*, pages 1–8, 2007.

[18] A. Shokoufandeh and S. J. Dickinson. A unified framework for indexing and matching hierarchical shape structures. In *4th Int. Workshop on Visual Form, LNCS 2059*, pages 67–84, 2001.

[19] R. Xu and I. Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005.