Median Graph Computation by means of a Genetic Approach Based on Minimum Common Supergraph and Maximum Common Subgraph

M. Ferrer[1], E. Valveny[2], and F. Serratosa[3]

[1] Institut de Robòtica i Informàtica Industrial, (UPC-CSIC), Spain
mferrer@iri.upc.edu
[2] Centre de Visió per Computador, Universitat Autònoma de Barcelona, Spain
ernest@cvc.uab.cat
[3] Departament d'Informàtica i Matemàtiques, Universitat Rovira i Virgili, Spain
francesc.serratosa@urv.cat

Abstract. Given a set of graphs, the median graph has been theoretically presented as a useful concept to infer a representative of the set. However, the computation of the median graph is a highly complex task and its practical application has been very limited up to now. In this work we present a new genetic algorithm for the median graph computation. A set of experiments on real data, where none of the existing algorithms for the median graph computation could be applied up to now due to their computational complexity, show that we obtain good approximations of the median graph. Finally, we use the median graph in a real nearest neighbour classification showing that it leaves the box of the only-theoretical concepts and demonstrating, from a practical point of view, that can be a useful tool to represent a set of graphs.

1 Introduction

In structural pattern recognition, the concept of median graph [1] has been presented as a useful tool to represent a set of graphs. Given a set of graphs S, the median graph is defined as the graph that minimizes the sum of distances (SOD) to all the graphs in S. It aims to extract the essential information of a set of graphs into a single prototype. Potential applications include graph clustering and prototype learning. For instance, it has been successfully applied to different areas such as image clustering [2], optical character recognition [1] and graphical symbol recognition [3].

Nevertheless, the computation of the median graph is a highly complex task. In the past some exact and approximate algorithms have been developed. Optimal algorithms include a tree search approach called multimatch [4] and a more efficient algorithm which takes advantage of certain conditions about the distance between two graphs [3] . Suboptimal methods include genetic algorithms [1], greedy-based algorithms [2] and spectral-based approaches such that of [3] and [5]. In spite of this wide offer of algorithmic tools, all of them are very limited in their application. They are often restricted to use small graphs and with some particular conditions. None of them have been applied using real data.

In this paper we tackle the problem of the median graph computation under a particular cost function. Using the results presented in [3], we present a new genetic algorihtm for the median graph computation that allows us to use a real database of 2,430 webpages. Firstly, we assess the accuracy of the median demonstrating that we are obtaining good approximations of the median graph. After that, we try to validate the median graph as a representative of a class of graphs. Up to now, existing algorithms could only be applied to very limited sets of graphs and the median graph could not be evaluated from a practical point of view as a good representative of a class. To that extent, we perform a preliminary classification experiment using the median graph. In some cases, we obtain slightly better results than a nearest-neighbor classifier with a much lower computation time. Thus, we demonstrate, for the first time, that the median graph can be a feasible alternative to represent a set of graphs in real applications.

The rest of the paper is organized as follows. In Section 2, we define the basic concepts used in the paper. Then, in Section 3 the concept of the median graph and its theoretical properties are presented. Section 4 introduces a new genetic algorithm for the median graph computation. Then, Section 5 is devoted to present our experiments and results. Finally, we terminate with some conclusions and possible future research lines.

2 Definitions and notation

Let L be a finite alphabet of labels for nodes and edges. A **graph** is a four-tuple $g = (V, E, \alpha, \beta)$ where V is the finite set of nodes, E is the set of edges, α is the node labelling function ($\alpha : V \longrightarrow L$), and β is the edge labelling function ($\beta : E \longrightarrow L$). We assume that our graphs are fully connected. Consequently, the set of *edges* is implicitly given (i.e. $E = V \times V$). Such assumption is only for notational convenience, and it does not impose any restriction in the generality of our results. In the case where no edge exists between two given nodes, we can include the special null label ε in the set of labels L to model such situation. Finally, the number of nodes of a graph g is denoted by $|g|$.

Given two graphs, $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$, g_2 is a **subgraph** of g_1, denoted by $g_2 \subseteq g_1$ if, $V_2 \subseteq V_1$, $\alpha_2(v) = \alpha_1(v)$ for all $v \in V_2$ and $\beta_2((u, v)) = \beta_1((u, v))$ for all $(u, v) \in V_2 \times V_1$. From this definition, it follows that, given a graph $g_1 = (V_1, E_1, \alpha_1, \beta_1)$, a subset $V_1 \subseteq V_1$ of its vertices uniquely defines a subgraph. Such subgraph is called the subgraph *induced* by V_2.

Let $S = \{g_1, g_2, ..., g_n\}$ be a set of graphs. A graph $g_m(S)$ is called a **maximum common subgraph of S** if $g_m(S)$ is a common subgraph of $\{g_1, g_2, \cdots, g_n\}$ and there is no other common subgraph of $\{g_1, g_2, \cdots, g_n\}$ having more nodes than $g_m(S)$. In addition, a graph $g_M(S)$ is called a **minimum common supergraph of S** if $\{g_1, g_2, \cdots, g_n\}$ are subgraphs of $g_M(S)$ and there is no other common supergraph of $\{g_1, g_2, \cdots, g_n\}$ having less nodes than $g_M(S)$. We will also denote the $g_m(S)$ and the $g_M(S)$ as $mcs(S)$ and $MCS(S)$ respectively.

The **graph edit distance** [6, 7] is commonly used to compute the dissimilarity between graphs. To that extent, a number of distortion or edit operations e, consisting of the insertion, deletion and substitution of both nodes and edges are defined. Then, for every pair of graphs (g_1 and g_2), there exists a sequence of edit operations, or edit path $p(g_1, g_2) = (e_1, \ldots, e_k)$ (e_i denotes an edit operation) that transforms one graph into the other. Several edit paths may exist between two graphs. This set of edit paths is denoted by $\wp(g_1, g_2)$. A cost c is assigned to each edit operation. The edit distance d between two graphs g_1 and g_2 denoted by $d(g_1, g_2)$ is the minimum cost edit path between two graphs.

In this work, we will use a particular cost function where the cost of node deletion and insertion is always 1, the cost of edge deletion and insertion is always 0 and the cost of node and edge substitution takes the values 0 or ∞ depending on whether the substitution is identical or not, respectively. Under this cost function, the edit distance between two graphs can be expressed as [8]:

$$d(g_1, g_2) = |g_1| + |g_2| - 2\,|mcs(g_1, g_2)| = |g_1| + |g_2| - 2\,|g_m| \qquad (1)$$

We will use Equation (1) as a distance measure in the rest of the paper.

3 Generalized Median Graph

Let U be the set of graphs that can be constructed using labels from L. Given $S = \{g_1, g_2, ..., g_n\} \subseteq U$, the **generalized median graph** $\bar{g}$ of S is defined as $g \in U$ that minimizes the sum of distances (SOD) to all the graphs in S:

$$\bar{g} = arg\ \min_{g \in U} \sum_{g_i \in S} d(g, g_i) = arg\ \min_{g \in U} SOD(g) \qquad (2)$$

Note that $\bar{g}$ is not usually a member of S and, in general, more than one generalized median graph can be found for a given set S.

The computation of the generalized median graph is a rather complex task. Both exact [4, 3] and approximate algorithms [1, 3, 5, 2] exist, although all of them can only be applied to very limited sets of graphs.

When the computation of the median graph is not possible, the set median graph $\hat{g}$ can be used. While the search space for $\bar{g}$ is U, the whole universe of graphs, the search space for $\hat{g}$ is simply S, the set of graphs in the learning set. The set median graph is usually not the best representative of a set of graphs, but it is often a good starting point towards the generalized median graph.

3.1 Theoretical Properties of the Median Graph

In [3], three new theoretical results, based on the use of the particular cost function presented in Section 2, are shown that can permit to reduce the computation time of the median graph. Basically these properties relate the search space, the size and the SOD of the median graph with the maximum common subgraph ($g_m(S)$) and the minimum common supergraph ($g_M(S)$) of the set of graphs:

– **Search space:** The search space of the median graph of S is composed only of all the induced subgraphs of $g_M(S)$.

– **Size of the median graph:** $0 \leq |g_m(S)| \leq |\bar{g}| \leq |g_M(S)| \leq \sum_{i=1}^{n} |g_i|$

– **Bounds of the SOD:** $SOD\,(\bar{g}) \leq SOD(g_m(S))$

In the following we will use these results to derive a new genetic algorithm for the median graph computation.

4 A New Genetic Algorithm

In this section we show how the three theoretical results presented in the previous section can be used to develop a new sub-optimal algorithm for the median graph computation. A genetic approach is chosen, since it allows us to easily encode a possible solution and also to explore the search space more efficiently.

In genetic algorithms [9], a possible solution of the problem is encoded using chromosomes, evaluated through a fitness function. Given an initial population of chromosomes, genetic operators, such as mutation or crossover, are applied to alter the chromosomes. This process is repeated until one or more stop conditions are satisfied. In the following we explain our particular implementation.

Chromosome Representation: From the property 1 of Section 2 it follows that a chromosome should be able to encode all the possible subgraphs of $g_M(S)$. Thus, the size of the chromosome is equal to the size of the $g_M(S)$. Each position in the chromosome, is associated to one node of $g_M(S)$, and may store either a value of "1" or a value of "0". Thus, the chromosome specifies which nodes of $g_M(S)$ are present (positions with "1") and which not (positions with "0"), generating an induced subgraph of $g_M(S)$, since a subset of nodes of a given graph uniquely defines a subgraph (Section 2). An example is shown in Figure 1. Assume that the $g_M(S)$ is the graph shown in Figure 1(a). the chromosome (Figure 1(c)) codifies the induced subgraph of $g_M(S)$ shown in Figure 1(b).

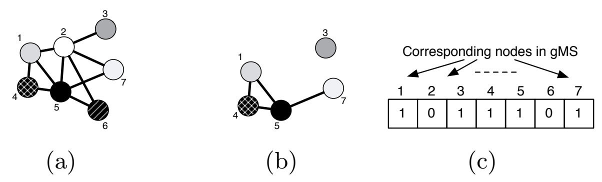

Fig. 1. A graph $g_M(S)$ (a), an induced subgraph g of $g_M(S)$ (b) and the chromosome representing g (c).

Fitness Function: The fitness function of each chromosome c corresponds to the SOD of the induced subgraph g of $g_M(S)$ the chromosome represents.

$$f(c) = SOD(g, S) = \sum_{i=1}^{n} d(g, g_i) = \sum_{i=1}^{n} (|g| - |g_i| - 2|mcs(g, g_i|) \qquad (3)$$

The lower its fitness function is, the better the chromosome is. This fitness function implies the computation of the maximum common subgraph of two graphs, which is exponential in the general case. Such computational complexity becomes polynomial when the considered graphs have unique node labels [10].

Genetic Operators: The crossover operator simply interchanges, with a uniform probability, an arbitrary position of two chromosomes. Mutation is accomplished by changing randomly a number in the array with a mutation probability. After the genetic operators have been applied, every chromosome is checked in order to validate whether it fulfils the bounds given in Section 3.1 regarding the size and the SOD of the median graph. If the chromosome is out of such limits, it is randomly altered until it fulfils the conditions. This procedure has two effects. First, we only take into account the induced subgraphs of $g_M(S)$ that fulfils the conditions. In addition, we expect to accelerate the convergence of the algorithm, since non-admissible candidate medians will never appear in the population. To create the descendants, a roulette wheel sampling implementing fitness-proportionate selection is chosen.

Population Initialization: The length of the initial population is set according to a predefined value K (20 in our case), determined empirically. Then, the first n chromosomes (with $n \leq K$) corresponds to the n graphs in S. In this way, we assure that the initial population includes the set median graph, which is a potential generalized median graph. The remaining K-n chromosomes are generated randomly but all of them must fulfil the bounds given in Section 3.

Termination Condition: The population evolution continues until either the maximum number of generations is reached or the best SOD in the population is less than the SOD of the set median graph.

5 Experiments

In this section we present two experiments on real data. The database is composed of 2,430 graphs containing 6 classes, with unique node labels, representing webpages with a mean size of around 200 nodes. In the first one, we evaluate the quality of the median graph according to the SOD. Finally, in a second experiment, we conduct a preliminary classification experiment in order to assess the median as a good representative of a given set of graphs.

Experiment 1. Median Accuracy: This experiment was intended to qualitatively evaluate the median graph computation achieved by the genetic approach. To this end, we computed the median graph of each class using 3, 4, 5, 6 and 7 graphs for each class, randomly selected. Then, we compare the SOD of the median computed using the genetic algorihtm with the SOD of the set median. We do not compare the results of the genetic approach with other methods, because none of them is able to deal with such large sets and graphs. This comparison can give a good idea of whether it is potentially a good median.

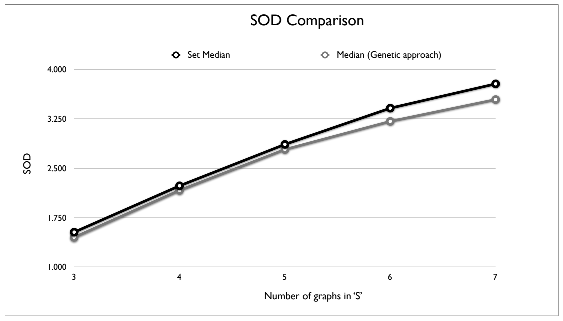

Fig. 2. SOD comparison function of the number of graphs in S.

Figure 2 show the results of this comparison as a function of the number of graphs in the set S. The results show that we obtain a better SOD with our method than with the set median for any number of graphs in the set. What is important in this figure is the tendency in the difference between the set median SOD and the SOD of the approximate median. This difference increases as the number of graphs in S increases. This tendency suggests that the more information of the class the method has (more elements in S), better representations is able to obtain.

With these results at hand, we can conclude that we obtain good approximations of the median graph with this new genetic approach.

Experiment 2. Classification Accuracy: In this experiment, the median graph is used to reduce (filter) the number of classes before the application of an 1NN classifier. That is, firstly, the median is used to obtain the representative of each class using the training set. Then, we compare every element of the test set against the median graph of all the classes. For each element in the test set, we rank all the classes according to the distance to these median graphs. After that, the same element is classified using the 1NN classifier but using only the elements in the training set of the best k classes according to the previous ranking, instead of using all the classes as in the 1NN classical approach. It is clear that, if k is set to 1, then the results are the same as those obtained with the classification using simply the median graph. Conversely, if k is equal to 6, then the results are the same as in the classical 1NN classifier. In order to better generalize the results, we performed 10 repetitions of the classification task. In each repetition, the training set was composed of 36 elements (6 per class), and the test set was composed by 324 elements (54 per class).

Figure 3 shows, for every value of k, the maximum, the average and the minimum classification accuracy achieved along the 10 repetitions of the experiment. An important observation is that, even for $k = 1$ or $k = 2$, the best results using the median graph could outperform the worse results using the 1NN classier. That means that, with the median graph, it is possible to achieve similar results to the 1NN, but using a lower number of comparisons.

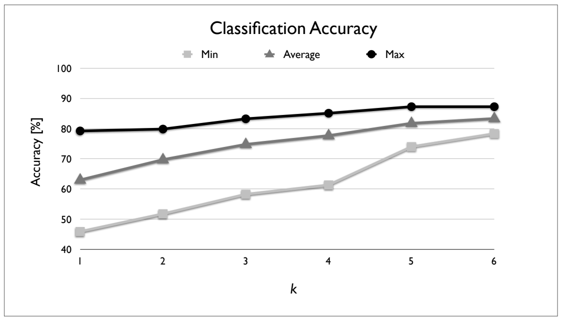

Fig. 3. Maximum, average and minimum classification accuracy for each value of k along the 10 repetitions.

Figure 4, shows for every value of k the number of repetitions where this value of k permits to obtain the same or better classification results as in the 1NN classifier. We can see that, in four repetitions (40%), we only need at most 4 classes ($k = 4$) to obtain better results. That means that, we can obtain the same calssification accuracy of the 1NN classifier with less number of comparisons (for $k = 4$ we need around a 24% less of comparisons than the 1NN classifier). In addition, in 80% of the repetitions, we need at most 5 classes to obtain better results than the 1NN classifier. In this case the reduction is around a 7%. Thus, the median graph can be used in this sense as a representative of a class.

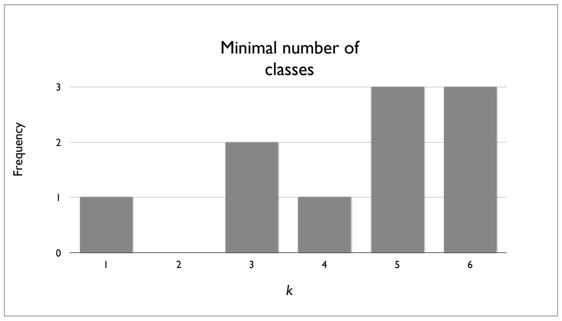

Fig. 4. Minimum number of classes to achieve the same results as in the 1NN classifier.

It is important to recall that this is a preliminary experiment with the aim to show that the median graph can be a good representative of a set of graphs. To apply the median graph to real classification problems, further work is required.

6 Conclusions and Future Work

The median graph has been presented as a good alternative to compute the representative of a set of graphs. The existing methods do not permit to use this concept it in real pattern recognition applications.

In this paper we have presented a new genetic algorithm for the median graph computation. With this new algorithm we performed a set of experiments using webpages extracted from real data. The first conclusion of these experiments is that, with this new algorithm, we are able to obtain accurate approximations of the median graph (in terms of SOD) with a computation time that permits to work with sets of graphs composed of around 200 nodes each. Although the applicability of the median graph to real problems is still limited, these results show that the concept of median graph can be used in real world applications. It demonstrates, for the first time, that the median graph is a feasible alternative to obtain a representative of a set of graphs. For instance, we have shown in a preliminary experiment that the classification using the median graph can obtain similar results as a nearest-neighbor classifier but with a lower computation time.

Nevertheless, there are still a number of issues to be investigated in the future. More accurate bounds or properties might be investigated using these new results in order to improve the knowledge of the median graph. Such advances may lead also to obtain more accurate and efficient approximate solutions of the median graph. Applying other optimization algorithms, such as tabu search, remains as an open path to be explored. In addition, the preliminary experiments on classification open the possibility of applying the median graph to classification algorithms where a representative of the set of graphs is required.

References

1. Jiang, X., Münger, A., Bunke, H.: On median graphs: Properties, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **23**(10) (2001) 1144–1151
2. Hlaoui, A., Wang, S.: Median graph computation for graph clustering. Soft Comput. **10**(1) (2006) 47–53
3. Ferrer, M.: Theory and algorithms on the median graph. application to graph-based classification and clustering. In: PhD Thesis, Universitat Autònoma de Barcelona. (2008)
4. Münger, A.: Synthesis of prototype graphs from sample graphs. In: Diploma Thesis, University of Bern (in German). (1998)
5. White, D., Wilson, R.C.: Mixing spectral representations of graphs. In: 18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China, IEEE Computer Society (2006) 140–144
6. Bunke, H., Allerman, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters **1**(4) (1983) 245–253
7. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man and Cybernetics **13**(3) (May 1983) 353–362
8. Bunke, H.: On a relation between graph edit distance and maximum common subgraph. Pattern Recognition Letters **18**(8) (1997) 689–694
9. Mitchel, M.: An Introduction to Genetic Algorithms. MIT Press. (1996)
10. Kandel, A., Bunke, H., Last, M.: Applied Graph Theory in Computer Vision and Pattern Recognition (Series in Computational Intelligence). Springer-Verlag, Berlin (2007)