A comparison between two representatives of a Set of Graphs: Median vs Barycenter Graph

Abstract. In this paper we consider two existing methods to generate a representative of a given set of graphs, that satisfy the following two conditions. On the one hand, that they are applicable to graphs with any kind of labels in nodes and edges and on the other hand, that they can handle relatively large amount of data. Namely, the approximated algorithms to compute the Median Graph via graph embedding and a new method to compute the Barycenter Graph. Our contribution is to give a new algorithm for the barycenter computation and to compare it to the median Graph. To compare these two representatives, we take into account algorithmic considerations and experimental results on the quality of the representative and its robustness, on several datasets.

1 Introduction

The straight advantages of the use of graphs for representation purposes appear to be useless in some applications due to the lack of mathematical structure in the graph domain. An illustrative example is the problem of finding a representative of a set of graphs. While in vector spaces it is easy to compute representatives such as medians and means with respect to a wide range of distances, in the graph domain the analogy turns out to be a highly non–trivial task.

In the literature we can distinguish different methodologies to tackle this problem, both probabilistic and deterministic. Random Graphs such as First-Order Random Graphs (FORGs) [20], Function-Described Graphs (FDGs) [17, 18] and Second-Order Random Graphs (SORGs) [16]; a Maximally General Prototype [4]; the Median Graph [9] and the Barycenter Graph [8] have been proposed as representatives of a set of graphs, among others. Most of these methods suffer from a prohibitive computation time or are limited to a restricted family of graphs.

In this paper we aim to compare those algorithms that on the one hand are applicable to graphs with any kind of labels in nodes and edges and on the other can handle relatively large amount of data. Namely, the approximated algorithms to compute the Median Graph via graph embedding and a new method to compute the Barycenter Graph. It includes some algorithmical considerations and experiments on several real–world and artificial datasets.

This paper is organized as follows. Some basic definitions are given in Section 2, the computation of the Median Graph is discussed in Section 3 and the proposed computation for the barycenter graph is presented in Section 4. Section 5 is devoted to comparing the Median and Barycenter Graphs. Finally, in Section 6 we draw some conclusions.

2 Definitions

Throughout the paper, let $S = \{g_1, g_2, ..., g_n\}$ be a set of graphs and let L be the set of labels of the nodes and edges of the graphs of S. Let U be the set of all graphs that can be constructed using labels from L, and observe that $S \subseteq U$. Also, let $d : U \times U \to \mathbb{R}$ be a distance over the set U.

Given a set of graphs S, the Set and Generalized Median Graphs [9] are defined as follows.

Definition 1 *The* set Median Graph $\hat{g}$ *and the* generalized Median Graph $\bar{g}$ *of* S *are defined as:*

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

The set Median Graph is a graph of the set S which minimizes the *sum of distances* (SOD) to all the graphs in S. The generalized Median Graph $\bar{g}$, is also a graph that minimizes the SOD to all the graphs in S, but the minimum is taken over U. Thus, the generalized Median Graph does not necessarily belong to the original set S. Since the minimum is taken over a larger set, the generalized Median Graph is expected to be a better representative for the set S of graphs. Notice that in general more than one set and generalized Median Graph may exist for a given set S.

Note that the median graph is analogous to the concept of median vector in a vector space. Similarly, the definition of Barycenter Graph is natural, by adapting the definition of barycenter of a set of points in $\mathbb{R}^n$.

Definition 2 *The* set Barycenter Graph $\hat{b}$ *and the* generalized Barycenter Graph *(or just* Barycenter Graph*)* $\bar{b}$ *of* S *are defined as:*

$$\hat{b} = arg \min_{g \in S} \sum_{g_i \in S} d(g, g_i)^2 \quad and \quad \bar{b} = arg \min_{g \in U} \sum_{g_i \in S} d(g, g_i)^2.$$

That is, the Barycenter Graph is the graph in U minimizing the *sum of squared distances* (SOSD) to all the graphs in S. The set barycenter is the argument minimizing the SOSD, when the search is limited to the given set S.

Although definitions 1 and 2 apply for any distance, we let d be the well known *graph edit distance* [15]. This choice makes it possible to apply the algorithms below to sets of graphs of different sizes and with any kind of labels.

Finally, we introduce the notion of *weighted mean*, first presented in [3].

Definition 3 *Let* g, g' *be graphs. Let* $I = \{h \in U \mid d(g, g') = d(g, h) + d(h, g')\}$*, be the set of* intermediate graphs*. Given* $0 \leq a \leq d(g, g')$*, the* weighted mean *of* g *and* g' *is a graph*

$$g'' = WM(g, g', a) = \arg\min_{h \in I} |d(g, h) - a|.$$

That is, given two graphs, g and g', and a parameter a, the weighted mean is an intermediate graph, not necessarily unique, whose distance to g is as similar as possible to a. Consequently, its distance to g' is also the closest to $d(g, g') - a$. Again, we let d be the graph edit distance.

Remark 1. Note that, the so called *error*, $\epsilon(a) = |d(g, g'') - a|$, is not necessarily null. This fact, regardless of the exactness of the computation, depends on the properties of the search space U.

3 Computation of the Median Graph

The most popular exact algorithm is called Multimatch [10], and was first presented by Münger and Bunke in 1995. This approach, as any exact Median Graph computation, suffers from a high computational complexity, and its application is very limited. The use of suboptimal methods is thus the unique feasible option to extend the use of the Median Graph to more realistic sets of graphs.

Approximate algorithms developed so far include a genetic based strategy [9, 10] and one greedy-based algorithm [7]. Both solutions generally apply some kind of heuristics in order to reduce both the cost of the graph distance computation and the size of the search space. Finally, the most recent approach, which is based on the proposal by Riesen et al [14], consists on embedding the graphs into an auxiliary Euclidean space. Let us give a more thorough explanation of this last technique, since the algorithms that we have used in the experiments presented in this paper follow it.

3.1 Median Graph via Graph Embedding

The general embedding procedure is composed of three main steps, detailed in the following.

- **Step I: Graph Embedding in a Vector Space:** Each graph in the set S is embedded into an n-dimensional vector space. The vector representation p_i of a graph g_i of the set is obtained by computing its distance to all the graphs in the set. More precisely, the j-th coordinate of the vector corresponds to the distance to the j-th graph of the set.
- **Step II: Median Vector Computation:** This step consists in computing the median vector $\bar{p}$ of the points obtained in the first step. Although the Euclidean Median cannot be calculated in a straightforward way [1], an approximation, as good as desired, can be obtained by means of the Weiszfeld's algorithm [19]. It is an iterative procedure that converges to the solution.
- **Step III: Going Back to the Graph Domain:** The last step consists in going back to the graph domain converting the median vector into a graph $\tilde{g}$. This graph is taken as the Median Graph of the set. Different options on how to perform this last step have been proposed.

Linear Interpolation Procedure. In this algorithm from [6], once the median vector $\bar{p}$, is computed, the two closest points, p_1 and p_2 without lost of generality, are used to obtain the approximate median. The approximate generalized Median Graph, $\tilde{g}$, is then the weighted mean of g_1 and g_2, with $a = \frac{1}{2}d(g_1, g_2)$. We will refer to this algorithm as *linear embedding* (**MLE**).

Triangulation Procedure. In this case, the three closest points to $\bar{p}$ are selected for the approximated generalized Median Graph computation. This computation consists on generating an intermediate weighted mean using two of the three points followed by a second and definitive weighted mean which makes use of the third point and the previous weighted mean. The procedure, referred to as *triangulation embedding* (**MTE**), was proposed and explained in [6].

Recursive Procedure. A third option is to take into account all the points, this is, all the graphs in the set S in Step III. That is what the authors propose in [5]. We will refer to this algorithm as *recursive embedding* (**MRE**).

4 Computation of the Barycenter Graph

The algorithm that we propose to approximate the Barycenter Graph is based on the following geometrical property of the barycenter in Euclidean spaces.

Lemma 4 *Given a set $P = \{p_1, p_2, \ldots, p_m\}$ of m points with $p_i \in \mathbb{R}^n$ for $i = 1 \ldots m$, the barycenter*

$$Bar(P) = arg \min_{y \in \mathbb{R}^n} \sum_{i=1}^{m} ||p_i - y||^2,$$

of the set P satisfies, for any $1 \leq j \leq m$,

$$Bar(P) = \frac{1}{m} p_j + \frac{m-1}{m} Bar(P \setminus \{p_j\}). \quad (1)$$

As it is deduced from equation (1), $Bar(P)$ lies in the segment with ends p_j and $Bar(P \setminus \{p_j\})$ and

$$\|Bar(P \setminus \{p_j\}) - Bar(P)\| = (m-1)\|Bar(P) - p_j\|,$$

where $\|\cdot\|$ denotes the Euclidean distance. Therefore, in Euclidean spaces, the barycenter of m points can be recursively computed by subtracting a point in the set and computing the barycenter of the remaining ones. Then, the barycenter is easy to compute because it belongs to a segment with known ends and the distance to these ends is also known.

4.1 Algorithm

The procedure explained above can be easily adapted to the domain of graphs, since the last step corresponds to the computation of the weighted mean. The resulting algorithm, **Algorithm 1**, may be considered an extension to the algorithm presented in [8]. The main contribution is that the graph edit distance, instead of the geometrically restricted distance function required in [8], can be used as the graph similarity measure of the graph domain.

Algorithm 1: Algorithm to approximate Barycenter Graph computation.

input : A set $S = \{g_1, \ldots, g_n\}$ of n graphs
output: $\tilde{b}$ = Approximate Barycenter Graph of S
begin
1 $B_2 = \mathrm{WM}(g_1, g_2, d(g_1, g_2)/2)$
2 **for** $3 \leq m \leq n$ **do**
3 $B_m = \mathrm{WM}(B_{m-1}, g_m, d(B_{m-1}, g_m)/m)$
4 Return $\tilde{b} = B_n$.

The output of **Algorithm 1** is an approximation $\tilde{b} \approx \bar{b}$ to the barycenter graph. This inaccuracy is on the one hand due to the error $\epsilon(a)$, and a consequence of the suboptimal computation of distances and weighted means, which is unavoidable unless the number and size of the graphs of the set S is very small. On the other hand, it cannot be theoretically proved than the algorithm minimizes the SOSD. Nevertheless, the fact that our method gives results with small SOSD is supported by experimental results.

It is important to remark that there is no need to transform the graphs into vectors to apply our method. This means that the structural information of the graphs is preserved at every step in the process.

4.2 Different sorting schemes

In **Algorithm 1** the graphs are taken as they arrive, without any sorting. Then the question whether the ordering of the input plays a non-negligible part in the accuracy of the approximation arises. For this reason, we have developed and implemented two methods, the *Ascendent SOSD–based* sorting (**BSA**) and the *Descendent SOSD–based* sorting (**BSD**), to study the effect of the ordering.

In the BSA method the graphs of the input are ordered upwards, such that the first graph, g_1, is that with minimum SOSD: the set barycenter. In the BSD method, the ordering of the graphs is the inverse.

The method explained in Section 4.1, without preprocessing the data, will be referred to as *unordered* barycenter computation method (**BN**). We also compute the set barycenter (**SB**).

5 Comparison Between Median and Barycenter Graphs

In this section we aim to compare the quality of the median and the Barycenter Graphs, as representatives of a set of graph. To do so, we compare experimental results on three real–data–based and five artificial datasets, some characteristics of which are displayed in Table 1. The LetterLOW, LetterHIGH, Molecules, Mutagenicity and Webpages datasets are from [12], where more information on them is available. The Synthetic datasets were created by the authors.

Table 1. Some dataset characteristics: size, number of classes (#c) and the average and maximum size of graphs.

Database	**Size**	**# c**	$\emptyset\|g\|$	**max**$\|g\|$	**Database**	**Size**	**# c**	$\emptyset\|g\|$	**max**$\|g\|$
LetterLOW	2,250	15	4.7	8	Webpages	2,340	6	186.1	834
LetterHIGH	2,250	15	4.7	8	SyntheticSmall	2,000	10	10	13
Molecules	2,000	2	15.7	95	SyntheticMedium	2,000	10	50	62
Mutagenicity	4,337	2	30.3	417	SyntheticLarge	2,000	10	100	122

The exact computation of both the generalized Median Graph and the generalized Barycenter Graph is unaffordable for these data. To carry on the experiments for this paper we have selected those suboptimal algorithms that can handle graphs with thousands of nodes. Table 2 shows the methods used.

Table 2. Methods to Approximate the Median and Barycenter Graphs and number of distances and weighted means that are computed for each of them, where n is the number of graphs in the given set S.

Method	**Shortening**	**#distances**	**#WMs**
Medians			
Set Median	SM	$\mathcal{O}(n^2)$	0
Linear Interpolation Embedding	MLE	$\mathcal{O}(n^2)$	1
Triangulation Embedding	MTE	$\mathcal{O}(n^2)$	2
Recursive Embedding	MRE	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Barycenters			
Set Barycenter	SB	$\mathcal{O}(n^2)$	0
Unordered Barycenter Computation	BN	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Ascendent SOSD–Based Sorting	BSA	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Descendent SOSD–Based Sorting	BSD	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

5.1 Algorithmical Considerations

Before showing the result of our experiments, we want to compare the different algorithms in terms of time complexity. Table 2 shows the number of distances

and weighted means that need to be computed for each of the methods, where n is the number of graphs in the given set S.

The embedding procedure has been shown to be the only method for the Generalized Median Graph computation potentially applicable to real world problems, due to its lower computational demand [6]. Still, the embedding step (Step I) requires the computation of all the pairwise distances of the graphs in the given set S. That means that the number of distances computed is quadratic on the number n of graphs. In terms of computation time Step II is negligible, and Step III does not depend on n if MLE or MTE are used. The MRE procedure computes a linear, on n, number of weighted means.

In all the barycenter computation algorithms from Table 2, a linear number of weighted means must be computed. In the case of BSA and BSD, the computation of a quadratic number of distances is also needed. The number of distance computations that requires the unsorted algorithm BN, is linear on n, instead.

In this paper, we have chosen to follow [11] and [13] for the graph edit distance computation and [3] to compute the weighted mean. This makes the graph edit distance computation more time demanding than the weighted mean, and the BN method the fastest one.

It is important to remark, then, that BN may be an interesting choice from the computational point of view. At sight of conclusions drawn in [2], the loss of quality of the approximation in comparison with other barycenter computations is small when special robustness against outliers is not needed. Finally, let us note that the BN method is incremental, making it unnecessary to store all the information to be processed. The rest of the algorithms are not.

5.2 Stability

Some of the methods that we are considering, namely MRE, BSA and BSD compute $n - 1$ intermediate approximations, being the last one taken as the definitive one. In this section we study, experimentally, the evolution of the quality of the approximation along these $n - 1$ steps. Recall that the Median Graph aims to minimize the SOD while the Barycenter Graph approximates the graph with minimum SOSD. For this reason, SOD and SOSD will be our reference values.

In this experiments we compute the Median Graph of several graph sets for letter, molecule, mutagenicity and web databases. More precisely, we compute the median and the barycenter of sets of 50 and 100 randomly chosen graphs belonging to the same class, and we do so for all the classes in each database and using each of the methods we want to evaluate. Each of these experiments is repeated 10 times.

As an example, Figure 1 shows the evolution of the SOD and SOSD, correspondingly, for the different methods in the experiments carried out with the Webpages dataset with sets of 50 graphs. We want to remark that the methods to compute the barycenter show a convergent tendency, while the evolution of the recursive embedding method is more irregular. Similar result concerning other datasets are skipped due to space constraints.

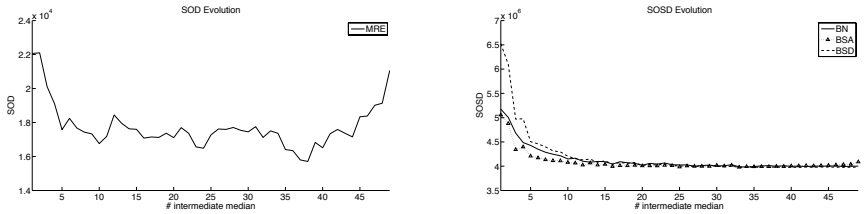

Fig. 1. Mean of the evolution of the SOD of the Median (left) and the SOSD of the Barycenter (right) for the **Web** dataset

5.3 Distance to prototype

In this section we present a second experiment, devoted to comparing the median and the barycenter as representatives of a given set of graphs. To this end, we have performed experiments with LetterLOW, LetterHIGH and the three Synthetic databases. All this datasets have been created by distorting initial prototypes. This allows us to compare the medians and barycenters provided by the different methods, to the original prototype. Under the assumption that the best possible representative is the prototype itself, we consider that the smaller the distance to the prototype, the better the representative is.

For each of the datasets, we have computed the representative of sets of different sizes (50 and 100 for letters and 10, 50 and 100 for Synthetics) using each of the methods displayed in Table 2. In each database, 20 sets of graphs are considered for each class. Figure 2 shows the mean distance of the resulting approximation to the prototype, taken over all the classes and all the repetitions.

In the LetterLOW and LetterHIGH datasets we observe that the set median, followed by the set barycenter is the closest representative to the prototype. Recall that, by definition, the generalized Median Graph has lower or equal SOD than the set median and similarly, the generalized barycenter has lower or equal SOSD than the set barycenter. This means that the set median and the set barycenter are expected to be worse representatives. In other words, they are the dummy approximations to beat.

We conclude that for the letter datasets, the approximated algorithms used for median and barycenter computation, although they have been experimentally validated [5, 6], fail to give satisfactory results. Let us remark that these databases suffer from a high level of distortion, potentiated by the fact that the graphs have few nodes. The embedding technique for Median Graph computation gives better representatives than the barycenter techniques. We may conclude that the Median Graph shows a higher robustness against large distortion. That it behaves better in difficult datasets, in other words.

In the Synthetic databases, the set median and the set barycenter are outperformed by all the algorithms to compute the generalized barycenter. Three facts are to be underlined. First, that the Barycenter Graphs give representatives closer to the prototype than the Median Graphs computed via embedding.

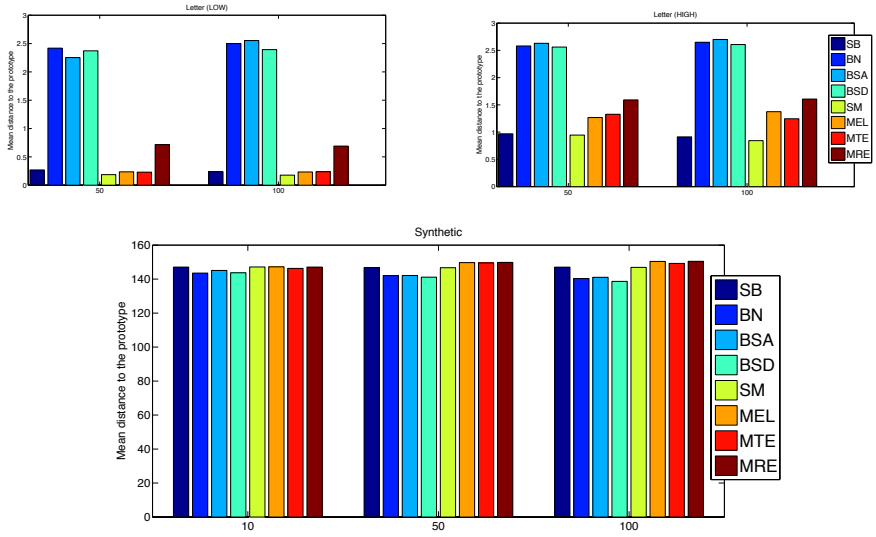

Fig. 2. Mean distance to prototype of several approximations to the barycenter and the median graphs, performed on letterLOW, letterHIGH, SyntheticSmall, SyntheticMedium and SyntheticLarge datasets.

Secondly, that the BSD gives the closest representatives to the prototype in the three Synthetic datasets. Thirdly, that the BN method, which as we said before is faster to compute than the rest of the methods, gives similar results to the rest of the barycenter methods.

6 Conclusions

In the present paper we have compared two representatives of a set of graphs, the median graph and the barycenter graph. Since their exact computation is unaffordable, this comparison is carried out by means of several algorithms that provide approximate medians and barycenters.

By comparing these algorithms we have concluded that an approximation to the barycenter can be computed faster than an approximation to the median. Also, we have noted that the algorithms for barycenters show a high level of convergence in the process of computing intermediate solutions, the last of which is the definitive approximation.

Finally, we have designed an experiment to discuss whether, among the median and the barycenter graph, one is better that the other as a representative. We have observed that results are not uniform for different datasets, which makes us conclude than none of them can be said to be better than the other. Nevertheless, we remark that, for datasets for which the grade of distorsion is not very high, barycenters give better representatives.

References

1. C. Bajaj. The algebraic degree of geometric optimization problems. *Discrete Comput. Geom.*, 3(2):177–191, 1988.
2. I. Bardaji. Graph representatives: Two different approaches based on the median and the barycenter graph. Master's thesis, UPC, Barcelona, 2009.
3. H. Bunke and S. Günter. Weighted mean of a pair of graphs. *Computing*, 67(3):209–224, 2001.
4. L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. Learning structural shape descriptions from examples. *Pattern Recognition Letters*, 23(12):1427–1437, 2002.
5. M. Ferrer, D. Karatzas, E. Valveny, and H. Bunke. A recursive embedding approach to median graph computation. In *7th IAPR-TC-15 International Workshop, GbRPR 2009, Venice, Italy, 2009, Proceedings*, LNCS. Springer, 2009.
6. M. Ferrer, E. Valveny, F. Serratosa, K. Riesen, and H. Bunke. Generalized median graph computation by means of graph embedding in vector spaces. *Pattern Recognition*, 43(4):1642 – 1655, 2010.
7. A. Hlaoui and S. Wang. Median graph computation for graph clustering. *Soft Comput.*, 10(1):47–53, 2006.
8. B. Jain and K. Obermayer. On the sample mean of graphs. In *Proc. of IJCNN 2008*, pages 993–1000, June 2008.
9. X. Jiang, A. Münger, and H. Bunke. On median graphs: Properties, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10):1144–1151, 2001.
10. A. Münger. Synthesis of prototype graphs from sample graphs. In *Diploma Thesis, University of Bern (in German)*, 1998.
11. M. Neuhaus, K. Riesen, and H. Bunke. Fast suboptimal algorithms for the computation of graph edit distance. In *Proc. of SSPR 2006. LNCS 4109*, pages 163–172, 2006.
12. K. Riesen and H. Bunke. IAM graph database repository for graph based pattern recognition and machine learning. In *SSPR/SPR*, pages 287–297, 2008.
13. K. Riesen, M. Neuhaus, and H. Bunke. Bipartite graph matching for computing the edit distance of graphs. In *Proc. of GbRPR 2007*, volume 4538 of *LNCS*, pages 1–12. Springer, 2007.
14. K. Riesen, M. Neuhaus, and H. Bunke. Graph embedding in vector spaces by means of prototype selection. In *6th IAPR-TC-15 International Workshop, GbRPR 2007*, volume 4538 of *LNCS*, pages 383–393. Springer, 2007.
15. A. Sanfeliu and K. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE TSMC*, 13(3):353–362, May 1983.
16. F. Serratosa, R. Alquézar, and A. Sanfeliu. Estimating the joint probability distribution of random vertices and arcs by means of second-order random graphs. In *Proc. of SSPR 2002 and SPR 2002, LNCS Vol. 2396*, pages 252–262, 2002.
17. F. Serratosa, R. Alquézar, and A. Sanfeliu. Synthesis of function-described graphs and clustering of attributed graphs. *IJPRAI*, 16(6):621–656, 2002.
18. F. Serratosa, R. Alquézar, and A. Sanfeliu. Function-described graphs for modelling objects represented by sets of attributed graphs. *Pattern Recognition*, 36(3):781–798, 2003.
19. E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Math. Journal*, (43):355– 386, 1937.
20. A. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE TPAMI*, 7:599–609, 1985.