Computing the Barycenter Graph by means of the Graph Edit Distance

Itziar Bardaji, Miquel Ferrer and Alberto Sanfeliu
Institut de Robòtica i Informàtica Industrial, UPC-CSIC, Spain
$\{$*ibardaji,mferrer,sanfeliu@iri.upc.edu*$\}$

Abstract

The barycenter graph has been shown as an alternative to obtain the representative of a given set of graphs. In this paper we propose an extension of the original algorithm which makes use of the graph edit distance in conjunction with the weighted mean of a pair of graphs. Our main contribution is that we can apply the method to attributed graphs with any kind of labels in both the nodes and the edges, equipped with a distance function less constrained than in previous approaches. Experiments done on four different datasets support the validity of the method giving good approximations of the barycenter graph.

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 graph domains. 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) [13], Function-Described Graphs (FDGs) [11, 12] and Second-Order Random Graphs (SORGs) [10]; a Maximally General Prototype [2] and the Set and Generalized Median graph [5] have been proposed as representatives of a set of graphs, among others. In this paper we use the so called *barycenter graph*, defined as the graph minimizing the sum of squared distances to a given set of graphs, as such a representative. One of its capital advantages agains the Median Graph and other representatives is that an incremental algorithm can be used for its approximation.

The barycenter graph was first defined in [3], where an algorithm for its computation was proved to be optimal for graph domains equiped with a geometrically constrained distance function. In this paper we propose a computation for the barycenter graph where the distance used in the graph domain is the graph edit distance. Our main contribution with respect to the previous work is that we are able to take profit from the flexibility of the graph edit distance, in constrast with the strong restriction on the distance assumed in [3]. As it is shown in this paper, our method has no limitation neither on the nature of the labels of the graphs nor on the cost function in which the graph edit distance is based. The validity of the algorithm, which is thus applicable to any set of graphs, is supported by experimental results on four different datasets.

This paper is organized as follows. Some basic definitions are given in Section 2, the computation proposed for the barycenter graph is presented in Section 3, Section 4 shows some experimental results and finally in Section 5 we draw some conclusions.

2. Definitions

To start with, let us formalize the concept of graph.

Definition 1 (Graph)**.** Given L, 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, $E \subseteq V \times V$ is the set of edges, α is the node labeling function ($\alpha : V \longrightarrow L$), and β is the edge labeling function ($\beta : E \longrightarrow L$).

The definition of barycenter graph is natural. We define it by adapting the definition of barycenter of a set of points in $\mathbb{R}^n$, which we recall in the following.

Definition 2 (Barycenter)**.** 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* or *centroid* is

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

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.241

where $\| \cdot \|$ denotes the Euclidean norm.

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. Also, let $d : U \times U \to \mathbb{R}$ be a distance over the set U.

Definition 3 (Barycenter Graph)**.** The *generalized barycenter graph* or *barycenter graph*, $\bar{b}$, of S is defined as:

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

This is, the barycenter graph is the graph in U minimizing the sum of squared distances (SOSD) to all the graphs in S. We can also define the set barycenter, which is the argument minimizing the SOSD, when the search is limited to the given set S itself.

Definition 4 (Set Barycenter Graph)**.** The *set barycenter graph* $\hat{b}$ of S is defined as:

$$\hat{b} = arg \min_{g \in S} \sum_{g_i \in S} d(g, g_i)^2. \tag{3}$$

Note that Definitions 3 and 4 are valid for any distance function d. In this paper we let d be the well known *graph edit distance* [9]. This choice makes it posible to apply the algorithms presented later to sets of graphs of different sizes and with any kind of labels.

Finally, we introduce the notion of *weighted mean*, first presented in [1], which plays a key role in the sequel. Let U and d be as before.

Definition 5 (Weighted Mean)**.** Let g, g' be graphs and

$$I = \{h \in U \mid d(g, g') = d(g, h) + d(h, g')\},$$

the set of *intermediate graphs*. Given $0 \leq a \leq d(g, g')$, the *weighted mean* of g and g' is a graph

$$g'' = \mathrm{WM}(g, g', a) = \arg\min_{h \in I} |d(g, h) - a|. \tag{4}$$

This 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, although the definition is valid for any distance function, we let d be the graph edit distance. For this distance function, an efficient computation of the weighted mean is given in [1]. Figure 1 shows an example of the weighted mean of a pair of graphs where the distance is a graph edit distance.

Remark 6. 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.

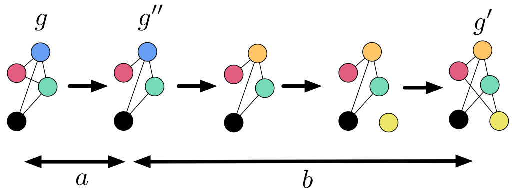

Figure 1. Example of a weighted mean. In this case a coincides with the deletion cost of the red–blue edge. Therefore, g'' is a WM for which $|d(g, g'') - a| = 0$.

3. Computation of Barycenter Graph

The algorithm that we propose for the approximated calculus of the barycenter graph is based on the following geometrical property of the barycenter in Euclidean spaces.

Lemma 7. *Given a set $P = \{p_1, p_2, \ldots, p_m\}$ of m points with $p_i \in \mathbb{R}^n$ for $i = 1 \ldots m$ and any $1 \leq j \leq m$, the barycenter of the set P satisfies*

$$Bar(P) = \frac{1}{m} p_j + \frac{m-1}{m} Bar(P \setminus \{p_j\}). \tag{5}$$

As it is deduced from equation (5), $Bar(P)$ 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 substracting 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.

3.1. Algorithm

The procedure explained above can be easily adapted to the domain of strings as in [4] and also to the domain of graphs, since the last step corresponds to the computation of the weighted mean. **Algorithm 1** results.

The output of **Algorithm 1** is an approximation $\tilde{b} \approx \bar{b}$ to the barycenter, but not equal to it in general. 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

Algorithm 1: Approximate Barycenter Graph

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

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.

Finally, let us note that **Algorithm 1** is incremental, making it unnecessary to store all the information to be processed.

3.2. Computing different sorting schemes

In **Algorithm 1** the graphs are taken as they arribe, 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 the following variants of the method, to study the effect of the ordering.

In the *Furthest–Ones–First* (**BF**) sorting scheme, we take g_1 and g_2 to be the pair of graphs with largest distance between them. Then B_2 is computed and g_3 is chosen to be the graph in $S \setminus \{g_1, g_2\}$ which is the furthest from B_2. Analogously, at each step the next graph introduced in the barycenter computation is the furthest from the previous intermediate barycenter. In the *Closest–Ones–First* sorting (**BC**), g_1 and g_2 are the two closest points of the set S and at each step the graph selected to be processed is the closest one to the intermediate barycenter. Finally, in the SOSD–based sortings, the graphs are ordered by SOSD. We present two methods, the *Ascendent SOSD–based* sorting (**BSA**) and the *Descendent SOSD–based* sorting (**BSD**). 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 3.1, without preprocessing the data, will be referred to as *unordered* barycenter computation method (**BN**). We also compute the set barycenter (**SB**).

4. Experimental Results

For the evaluation of the different methods to compute the barycenter graph proposed in Section 3, we have used four different databases from [7]: letter, molecule, mutagenicity and web databases. Some characteristic of these datasets are shown in Table 1.

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

Database	Size	# classes	$\emptyset\|g\|$	max$\|g\|$
Letter	2,250	15	4.7	8
Molecules	2,000	2	15.7	95
Mutagenicity	4,337	2	30.3	417
Webpages	2,340	6	186.1	834

In this experiments we compute the barycenter graph of several graph sets for each of the databases. More precisely, we compute the barycenter graph 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. Each of these experiments is repeated 10 times. We also compute the set barycenter for each of these sets.

We follow [6] and [8] for the graph edit distance computation and [1] to compute the weighted mean.

For each database and each method, Figure 2 shows the mean of the SOSD of the barycenter to the input set, taken over all the classes and the ten repetitions performed for each class. The lower the SOSD, the better the barycenter found.

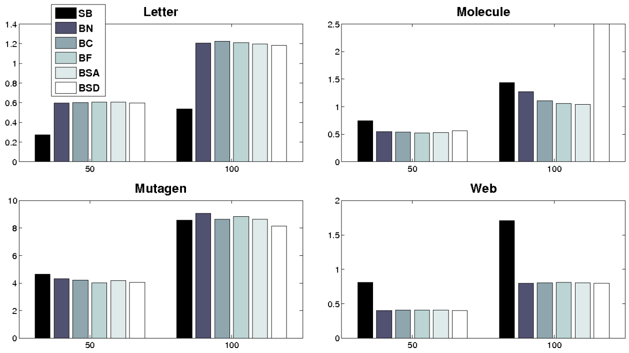

Figure 2. Mean SOSD of the barycenters to the set, in four databases

Note that, trivially, the SOSD of the set barycenter is an upper bound of the SOSD of the barycenter graph. Since the exact computation of the set barycenter is afordable, for a generalized barycenter graph to be acceptable, its SOSD to the set must be smaller than the SOSD of the set barycenter. Unfortunately, we do not have a lower bound for the SOSD of the true barycenter graph.

As it can be seen in Figure 2, in the web database all our methods give substancially better results than the set barycenter (in dark blue). In the mutagenicity database all the methods outperform the set barycenter in experiments with 50 graph and for larger sets BSD does. For molecules only one of the barycenter graph aproximation gives higher SOSD than the set barycenter. These results show that our algorithms return graphs satisfactorily similar to the barycenter graph.

Nevertheless, at sight of the result for the letter database, we conclude that our method is not to be used universaly and that there are datasets for which it does not work well. Which are the properties behind this behaviour is a nice question to ask in the future.

With respect to the different variant methods, presented in Section 3.2, we remark that BSD gives the best results in 5 out of the 8 instances of the experiments. Thevertheless, BF and BSA also give good numbers and behave more regularly in all the datasets.

The high value of SOSD given by the BSD method in the molecule dataset is an interesting fact. In the BSD method, the first graphs taken into account for the calculus are those with higher SOSD to the set. This means that if there are outliers, then they are the first graphs in the ordering. We conjecture that the first graphs in the ordering have a stronger influence in the final result than the last ones and that, for this reason, the BSD method is specially sensitive to outliers and BSA is not.

Finally we make note that the BN method requires the computation of $n-1$ distances and weighted means, while the rest of the methods require the computation of the whole distance matrix for the preprocessing, which involves a quadratic, instead or linear, number of distances. In most of the experiments shown in Figure 2, BN gives reasonable results. Thus, this may be the most interesting method if the input set is large.

5. Conclusions

In this paper we have presented a new method to compute a representative of set of graphs. More precisely, we approximate the barycenter graph: the argument minimizing the SOSD to the given set of graphs.

We have performed experiments on four different databases using 5 different algorithms based on this method. Results show that in most instances the SOSD of the graph obtained is lower than the upper bound to beat, given by the set barycenter graph. Also, we have concluded that preprocessing the input data is not advisable when the sets of graphs are large. However, it may be interesting to consider this prepocesing if robustness agains outliers is needed. The study of this last fact will be considered in the future.

Acknowledgments

This work has been supported by the Spanish research programmes Consolider Ingenio 2010 CSD2007-00018.

References

[1] H. Bunke and S. Günter. Weighted mean of a pair of graphs. *Computing*, 67(3):209–224, 2001.

[2] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. Learning structural shape descriptions from examples. *Pattern Recognition Letters*, 23(12):1427–1437, 2002.

[3] B. Jain and K. Obermayer. On the sample mean of graphs. In *Proc. of IJCNN 2008*, pages 993–1000.

[4] X. Jiang, K. Abegglen, H. Bunke, and J. Csirik. Dynamic computation of generalised median strings. *Pattern Anal. Appl.*, 6(3):185–193, 2003.

[5] 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.

[6] 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.

[7] K. Riesen and H. Bunke. IAM graph database repository for graph based pattern recognition and machine learning. In *SSPR/SPR*, pages 287–297, 2008.

[8] 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.

[9] A. Sanfeliu and K. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE TSMC*, 13(3):353–362, May 1983.

[10] 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.

[11] F. Serratosa, R. Alquézar, and A. Sanfeliu. Synthesis of function-described graphs and clustering of attributed graphs. *IJPRAI*, 16(6):621–656, 2002.

[12] 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.

[13] A. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE TPAMI*, 7:599–609, 1985.