# Graph edit distance or graph edit pseudo-distance?

SERRATOSA, F., CORTÉS, X., and MORENO, C.

2016

# Graph Edit Distance or Graph Edit Pseudo-Distance?

Francesc Serratosa
Universitat Rovira i Virgili
Tarragona, Catalonia, Spain
francesc.serratosa@urv.cat

Carlos Francisco Moreno-García
Universitat Rovira i Virgili
Tarragona, Catalonia, Spain
carlsofrancisco.moreno@urv.cat

Xavier Cortés
Universitat Rovira I Virgili
Tarragona, Catalonia, Spain
xavier.cortes@.urv.cat

***Abstract*— Graph Edit Distance has been intensively used since its appearance in 1983. This distance is really useful if we want to compare a pair of attributed graph from any domain and obtain not only a distance, but also the best correspondence between nodes of the involved graphs. A lot of efforts have been made to define fast and accurate optimal or sub-optimal error-tolerant graph matching algorithms, since it is known that the exact computation of the Graph Edit Distance has an exponential computational cost. In this paper, we want to analyse if the Graph Edit Distance can be really considered a distance or a pseudo-distance, since some restrictions of the distance function are not fulfilled. Distinguishing between both cases is important because being a distance is a restriction in some methods to return exact instead of approximate results. For instance, it happens in some graph retrieval techniques. Experimental validation shows us that in most of the cases, it is not correct to denominate it a distance, but a pseudo-distance instead, since the triangle inequality is not fulfilled. Therefore, in these cases, the graph retrieval techniques not always return the optimal graph.**

***Keywords: Graph Edit Distance, Edit Cost, Distance function.***

## I. INTRODUCTION

Attributed graphs have been of crucial importance in pattern recognition throughout more than four decades [1], [2]. They have been used to model several kinds of problems in some pattern recognition fields such as object recognition, scene view alignment, multiple object alignment, object characterization, among a great amount of other applications. Interesting reviews of techniques and applications are [3], [4] and [5]. If elements in pattern recognition are modelled through attributed graphs, error-tolerant graph-matching algorithms are needed that aim to compute a matching between nodes of two attributed graphs that minimizes some kind of objective function. To that aim, one of the most widely used methods to evaluate an error correcting graph isomorphism is the Graph Edit Distance [1], [2], [6].

Graph Edit Distance needs two main input parameters, which are the pair of attributed graphs to be compared and also other calibration parameters. These parameters have to be tuned to maximise a recognition ratio in a classification scenario or simply to minimise the Hamming distance between a ground-truth correspondence between nodes of both graphs and the obtained correspondence. It turns out that little research has been done to analyse if really the Graph Edit Distance is a distance or simply a similarity function that could be classified as a pseudo-distance, since some distance restrictions are not fulfilled. Reference [7] is the only paper related on this idea, and it shows in which conditions of these calibration parameters the Graph Edit Distance is really a distance.

The importance to the Graph Edit Distance being a true distance has an influence on some applications. As an example, in [8], [9] and [10] they present a method to retrieve graphs in a database. They suppose that given three graphs, the triangle inequality is fulfilled and thanks to this assumption, some comparisons were not needed to be performed. It turns out that if the Graph Edit Distance is not a distance, then the triangle inequality is not guaranteed, and then some graphs that would have to be explored are not considered, making the method to become sub-optimal.

The aim of this paper is to empirically analyse if the cases that the recognition ratio is maximised or the Hamming distance between the ground truth and the obtained correspondence are minimised are the ones in which the restrictions between parameters imposed by the distance definition are hold.

The outline of the paper is as follows; in sections 2 and 3, we define the attributed graphs and the Graph Edit Distance. In sections 4 and 5, we explain the restrictions needed to be a function a distance and we relate these restrictions on the specific case of the Graph Edit Distance. In Section 5, we show the experimental validation to deduct the parameters that maximise the classification ratio or minimise the Hamming distance. Finally, Section 6 concludes the paper.

## II. GRAPH & CORRESPONDENCE BETWEEN GRAPHS

Let $\Delta_v$ and $\Delta_e$ denote the domains of possible values for attributed vertices and arcs, respectively. An attributed graph (over $\Delta_v$ and $\Delta_e$) is defined by a tuple $G = (\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v = \{v_k \mid k = 1, \dots, R\}$ is the set of vertices (or nodes), $\Sigma_e = \{e_{ij} | i, j \in \{1, \dots, R\}\}$ is the set of edges (or arcs), $\gamma_v: \Sigma_v \rightarrow \Delta_v$ assigns attribute values to vertices and $\gamma_e: \Sigma_e \rightarrow \Delta_e$ assigns attribute values to edges.

Let $G^p = (\Sigma_v^p, \Sigma_e^p, \gamma_v^p, \gamma_e^p)$ and $G^q = (\Sigma_v^q, \Sigma_e^q, \gamma_v^q, \gamma_e^q)$ be two attributed graphs of order $R^p$ and $R^q$. To allow maximum flexibility in the matching process, graphs can be extended with null nodes [1] to be of order $R^p + R^q$. We refer to null nodes of $G^p$ and $G^q$ by $\hat{\Sigma}_v^p \subseteq \Sigma_v^p$ and $\hat{\Sigma}_v^q \subseteq \Sigma_v^q$ respectively. Let T be a set of all possible correspondences between two vertex sets $\Sigma_v^p$ and $\Sigma_v^q$. Correspondence $f^{p,q}: \Sigma_v^p \rightarrow \Sigma_v^q$, assigns each vertex of $G^p$ to only one vertex of $G^q$. The correspondence

between edges, denoted by $f_e^{p,q}$, is defined accordingly to the correspondence of their terminal nodes.

$$f_e^{p,q}\left(e_{ab}^p\right) = e_{ij}^q \Rightarrow f^{p,q}\left(v_a^p\right) = v_i^q \wedge f^{p,q}\left(v_b^p\right) = v_j^q \quad (1)$$

$$v_a^p, v_b^p \in \Sigma_v^p - \hat{\Sigma}_v^p \text{ and } v_i^q, v_j^q \in \Sigma_v^q - \hat{\Sigma}_v^q$$

We define non-existent or null edges by $\hat{\Sigma}_e^p \subseteq \Sigma_e^p$ and $\hat{\Sigma}_e^q \subseteq \Sigma_e^q$.

## III. Graph Edit Distance

The basic idea behind the Graph Edit Distance is to define a dissimilarity measure between two graphs. This dissimilarity is defined as the minimum amount of distortion required to transform one graph into the other. To this end, a number of distortion or edit operations, consisting of insertion, deletion and substitution of both nodes and edges are defined. Then, for every pair of graphs ($G^p$ and $G^q$), there is a sequence of edit operations, or an edit path $\text{editPath}(G^p, G^q) = (\varepsilon_1, \ldots, \varepsilon_k)$ (where each $\varepsilon_i$ denotes an edit operation) that transforms one graph into the other. In general, several edit paths may exist between two given graphs. This set of edit paths is denoted by $\vartheta$. To quantitatively evaluate which edit path is the best, edit cost functions are introduced. The basic idea is to assign a penalty cost to each edit operation according to the amount of distortion that it introduces in the transformation.

Each $\text{editPath}(G^p, G^q) \in \vartheta$ can be related to an univocal correspondence $f^{p,q} \in T$ between the involved graphs. This way, each edit operation assigns a node of the first graph to a node of the second graph. Deletion and insertion operations are transformed to assignations of a non-null node of the first or second graph to a null node of the second and first graph. Substitutions simply indicate node-to-node assignations. Using this transformation, given two graphs, $G^p$ and $G^q$, and a correspondence between their nodes, $f^{p,q}$, the graph edit cost is given by [1]:

$$\begin{aligned} EditCost(G^p, G^q, f^{p,q}) = \\ \sum_{\substack{v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \\ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q}} C_{ns}\left(v_a^p, v_i^q\right) + \sum_{\substack{e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \\ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q}} C_{es}\left(e_{ab}^p, e_{ij}^q\right) + \\ \sum_{\substack{v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \\ v_i^q \in \hat{\Sigma}_v^q}} C_{nd}\left(v_a^p, v_i^q\right) + \sum_{\substack{v_a^p \in \hat{\Sigma}_v^p \\ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q}} C_{ni}\left(v_a^p, v_i^q\right) + \\ \sum_{\substack{e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \\ e_{ij}^q \in \hat{\Sigma}_e^q}} C_{ed}\left(e_{ab}^p, e_{ij}^q\right) + \sum_{\substack{e_{ab}^p \in \hat{\Sigma}_e^p \\ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q}} C_{ei}\left(e_{ab}^p, e_{ij}^q\right) \end{aligned} \quad (2)$$

$$\text{being } f^{p,q}\left(v_a^p\right) = v_i^q \text{ and } f_e^{p,q}\left(e_{ai}^p\right) = e_{ij}^q$$

where $C_{ns}$ is the cost of substituting node $v_a^p$ of $G^p$ by node $f^{p,q}\left(v_a^p\right)$ of $G^q$, $C_{nd}$ is the cost of deleting node $v_a^p$ of $G^p$ and $C_{ni}$ is the cost of inserting node $v_i^q$ of $G^q$. Equivalently for edges, $C_{es}$ is the cost of substituting edge $e_{ab}^p$ of graph $G^p$ by edge $f_e^{p,q}\left(e_{ab}^p\right)$ of $G^q$, $C_{ed}$ is the cost of assigning edge $e_{ab}^p$ of $G^p$ to a non-existing edge of $G^q$ and $C_{ei}$ is the cost of assigning edge $e_{ab}^q$ of $G^q$ to a non-existing edge of $G^p$.

Finally, the Graph Edit Distance is defined as the minimum cost under any correspondence in T:

$$GED(G^p, G^q) = \min_{f^{p,q} \in T} EditCost(G^p, G^q, f^{p,q}) \quad (3)$$

Using this definition, the Graph Edit Distance essentially depends on $C_{ns}, C_{nd}, C_{ni}, C_{es}, C_{ed}$ and $C_{ei}$ functions. Several definitions of these functions exist. Table 1 summarises the five different configurations presented until today.

The first option [11], [12], [13], [14] are the ones where the whole costs are defined as functions that depend on the involved attributes and also on other learned or general knowledge. Attributes are density functions instead of vectors of attributes. The second option makes the Graph Edit Distance to be directly related to the maximal common sub-graph. That is, in [15], authors demonstrate that computing the Graph Edit Distance is exactly the same than deducting the maximal common sub-graph. In the third option, [16], authors assume that the graphs are complete, and a non-existing edge is an edge with a "null" attribute. In this case, the cost of deleting and inserting an edge is encoded in the edge substitution cost. Inserting and deleting nodes have a constant cost, $K_n$. With this definition, authors describe several classes of costs that equation 3 deducts the same correspondence. The fourth option might be the most used one [1], [17], [18]. Substitution costs are defined as distances between vectors of attributes, usually the Euclidean distance. Insertion and deletion costs are constants, $K_n$ and $K_e$, that have been manually tested or automatically learned [19], [20]. Finally, the last option is used in fingerprint recognition [21]. It is similar to the previous option, except from the substitution costs that are constants. Nodes represent minutiae and edges are the relations between them. If a specific distance between minutiae is lower than a threshold, then a zero is imposed as a substitution cost. Otherwise, this cost takes a constant value $K_{ns}$. The same happens with the edges that take a constant value $K_{es}$.

TABLE I. Examples of graph edit costs

| Ref. | $C_{ns}$ | $C_{nd}$ | $C_{ni}$ | $C_{es}$ | $C_{ed}$ | $C_{ei}$ |
|---|---|---|---|---|---|---|
| [11] [12] [13] [14] | $d_n(v_a^p, v_i^q)$ | $f_{nd}(v_a^p)$ | $f_{nd}(v_i^q)$ | $d_e(e_{ab}^p, e_{ij}^q)$ | $f_{ed}(e_{ab}^p)$ | $f_{ei}(e_{ij}^q)$ |
| [15] | $0, \infty$ | 1 | 1 | $0, \infty$ | 0 | 0 |
| [16] | $d_n(v_a^p, v_i^q)$ | $K_n$ | $K_n$ | $d_e(e_{ab}^p, e_{ij}^q)$ | 0 | 0 |
| [1] [17] [18] | $d_n(v_a^p, v_i^q)$ | $K_n$ | $K_n$ | $d_e(e_{ab}^p, e_{ij}^q)$ | $K_e$ | $K_e$ |
| [21] | $0, K_{ns}$ | $K_n$ | $K_n$ | $0, K_{es}$ | $K_e$ | $K_e$ |

It is worth noting that all of the cases, except for the first one, the insertion and deletion costs on nodes are considered to be the same, $K_n$. The same happens for edges, $K_e$. Nevertheless, in the string edit distance, also known as Levenshtein distance [22], insertion and deletion costs might

be considered different depending on the application. The most usual application is an automatic writing correction, in which the possibility of missing a character is different than accidentally adding an extra character [23].

The optimal computation of the Graph Edit Distance is usually carried out by means of a tree search algorithm, which explores the space of all possible mappings of the nodes and edges of the first graph to the nodes and edges of the second graph. A widely used method is based on the A* algorithm, for instance [18]. Unfortunately, the computational complexity of this algorithm, although a heuristic function can be used to reduce the space search, is exponential in the number of nodes of the involved graphs. This means that the running time may be non-admissible in some applications, even for reasonably small graphs. This is why Bipartite graph matching [24], [25] has appeared o be one of the newest methods presented to solve the Graph Edit Distance in a sub-optimal way. Experimental validation shows that, nowadays, it is one of the best sub-optimal algorithms since it obtains a good approximation of the distance in cubic computational cost. Interesting surveys on graph matching are [3], [4] and [5].

## IV. Defining the Graph Edit Distance as a True Distance

A distance, also called a metric, is a function that defines a dissimilarity between elements of a set. The domain is $[0, \infty)$ and it holds the following restrictions for all elements in the set [26]:

1) Non-negativity: $dist(x, y) \geq 0$.
2) Identity of indiscernibles: $dist(x, y) = 0 \Leftrightarrow x = y$.
3) Symmetry: $dist(x, y) = dist(y, x)$ (4)
4) Triangle inequality: $dist(x, y) \leq dist(x, z) + dist(x, z)$

In some cases, it is needed to relax these restrictions and the resulting functions are not called distance but pseudo-distance, quasi-distance, meta-distance or semi-distance, depending on which restriction is violated and how it is violated [26].

All in all, and independently of the definition of the edit costs, it was demonstrated in [7] that if we wish the Graph Edit Distance to be defined as a true distance function, it is needed to assure the whole edit operations in the edit path used to deduct the final distance (equation 3) fulfil the four properties in the following equation 5. In these equations, we suppose that the edit path generates correspondence $f^{p,q}$ such that $f^{p,q}(v_a^p) = v_i^q$ and $f^{p,q}(v_b^p) = v_j^q$.

1) Non-negativity: $\mathrm{C}_{\mathrm{ns}} \geq 0$ and $\mathrm{C}_{\mathrm{es}} \geq 0$ .
2) Identity of indiscernibles:
$C_{ns}(v_a^p, v_i^q) = 0 \Leftrightarrow \gamma_{\mathrm{v}}(v_a^p) = \gamma_{\mathrm{v}}(v_i^q)$
$C_{es}(e_{ab}^p, e_{ij}^q) = 0 \Leftrightarrow \gamma_{\mathrm{e}}(e_{ab}^p) = \gamma_{\mathrm{e}}(e_{ij}^q)$
3) Symmetry:
$\mathrm{C}_{\mathrm{nd}}(v_a^p, v_i^q) = \mathrm{C}_{\mathrm{ni}}(v_{a'}^p, v_{i'}^q) \Leftrightarrow \gamma_{\mathrm{v}}(v_a^p) = \gamma_{\mathrm{v}}(v_{i'}^q)$ (5)
where $v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p$, $v_i^q \in \hat{\Sigma}_v^q$, $v_{a'}^p \in \hat{\Sigma}_v^p$ and $v_{i'}^q \in \Sigma_v^q - \hat{\Sigma}_v^q$
$\mathrm{C}_{\mathrm{ed}}(e_{ab}^p, e_{ij}^q) = \mathrm{C}_{\mathrm{ei}}(e_{a'b'}^p, e_{i'j'}^q) \Leftrightarrow \gamma_{\mathrm{e}}(e_{ab}^p) = \gamma_{\mathrm{e}}(e_{i'j'}^q)$
where $e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p$, $e_{ij}^q \in \hat{\Sigma}_e^q$, $e_{a'b'}^p \in \hat{\Sigma}_e^p$ and $e_{i'j'}^q \in \Sigma_e^q - \hat{\Sigma}_e^q$
4) Triangle inequality:
$\mathrm{C}_{\mathrm{ns}}(v_a^p, v_i^q) \leq \mathrm{C}_{\mathrm{nd}}(v_a^p, v_{i'}^q) + \mathrm{C}_{\mathrm{ni}}(v_{a'}^p, v_i^q)$
where $v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p$, $v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q$ $v_{i'}^q \in \hat{\Sigma}_v^q$ and $v_{a'}^p \in \hat{\Sigma}_v^p$
$\mathrm{C}_{\mathrm{es}}(e_{ab}^p, e_{ij}^q) \leq \mathrm{C}_{\mathrm{ed}}(e_{ab}^p, e_{i'j'}^q) + \mathrm{C}_{\mathrm{ei}}(e_{a'b'}^p, e_{ij}^q)$
where $e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p$, $e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q$ $e_{i'j'}^q \in \hat{\Sigma}_e^q$ and $e_{a'b'}^p \in \hat{\Sigma}_e^p$

For all cited references, functions in table 1 are defined as distances, and constants as real positive numbers. For this reason, if the Graph Edit Distance cannot be defined as a true distance, it is due to the relations between these functions and constants. Considering the five options proposed in table 1, we realise that the second and third ones do not hold the triangle inequality and therefore cannot be considered as distances. It is really difficult to analyse the first option since being a distance or not depends on the specific distance values. The fourth option is a distance only if it is guaranteed that the whole substitution operations in the edit path hold:

$$d_n(v_a^p, v_i^q) \leq 2 \cdot K_n \text{ and } d_e(e_{ab}^p, e_{ij}^q) \leq 2 \cdot K_e \quad (6)$$

That is, we only have to analyse if the triangle inequality of equation 5 is fulfilled. Finally, the last option is almost the same than the third one and it is a true distance if constant costs are defined such that,

$$K_{ns} \leq 2 \cdot K_n \text{ and } K_{es} \leq 2 \cdot K_e \quad (7)$$

Since the fourth option is both the most used and the one that can be defined as distance or not, depending on the costs, from now on, we concretise on this specific case.

## V. Deducting the Edit Costs through a Ground Truth Correspondence

Note that given a pair of graphs and an optimal correspondence (the one that minimise $EditCost$ in equation 3), we can analyse if the used edit costs make the Graph Edit Disstance to be a true distance or not. Moreover, each combination of edit costs generates a different optimal correspondence and a Graph Edit Distance value. For this reason, the problem of knowing which are the edit costs that make the Graph Edit Distance to be a true distance is a chicken egg problem. Given some edit costs, we need to compute the optimal correspondence to deduct if the four distance restrictions are violated (equation 5), but to deduct the proper edit costs, we need the optimal correspondence.

To solve this problem, we propose to use a ground truth correspondence. That is, given a pair of attributed graphs, and independently of the edit costs, a human or another method deducts which is the "best" correspondence. Thus, we consider that the Graph Edit Distance is a true distance if the four properties in equation 5 are fulfilled assuming that $f^{p,q}$ in equation 5 is the ground truth correspondence.

Given an application that involves an attributed graph database of $M$ graphs in which the computation of the Graph Edit Distance is needed, the same edit costs have to be used in the whole process and graphs. Thus, we generalise equation 6 considering that we have several graphs and also introducing the ground truth concept. We conclude that the Graph Edit Distance is a true distance given some specific insertion and deletion costs for nodes if the following equation holds,

$$\begin{aligned} &\forall\, v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \text{ given } p\colon 1..M \text{ and } a\colon 1..(\mathrm{R^p} + \mathrm{R^q}) \\ &\text{such that } f^{p,q}\left(v_a^p\right) = v_i^q \;\&\; v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \\ &\text{leads to } d_n\left(v_a^p, v_i^q\right) \leq 2 \cdot K_n \\ &\text{being } f^{p,q} \text{ the ground-truth correspondence} \end{aligned} \tag{8}$$

Similarly happens for the edges,

$$\begin{aligned} &\forall\, e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \text{ given } p\colon 1..M \text{ and } a,b\colon 1..(\mathrm{R^p} + \\ &\mathrm{R^q}) \text{ such that } \mathrm{f_e^{p,q}}\left(e_{ab}^p\right) = e_{ij}^q \;\&\; e_i^{jq} \in \Sigma_e^q - \hat{\Sigma}_e^q \\ &\text{leads to } d_e\left(e_{ab}^p, e_{ij}^q\right) \leq 2 \cdot K_e \\ &\text{being } \mathrm{f_e^{p,q}} \text{ the edge correspondence deducted from the} \\ &\text{ground-truth correspondence } f^{p,q}. \end{aligned} \tag{9}$$

In the next section we empirically test if the costs that obtain the best recognition ration and the minimum Hamming distance between the ground truth correspondence and the obtained correspondence make the Graph Edit Distance a true distance or only a pseudo-distance since the triangle inequality is not hold.

## VI. Experimentation

We used five graph databases that are organised in registers such that each register is composed of a pair of graphs and a ground truth correspondence between their nodes. These databases were initially used to automatically learn insertion and deletion edit costs in [19] and [20], and are publically available in [27]. These databases do not have attributes on the edges and therefore, we only analyse the insertion and deletion costs on nodes. Nonetheless, what can be deducted on nodes could be easily extrapolated to edges. Graphs in the first three databases, *Letter Low*, *Letter Med* and *Letter High*, represent hand written characters, which nodes have as only attribute the (x,y) position of the junctions of strokes in the character, and edges being the strokes. Graphs in *House-Hotel* database and *Tarragona RotationZoom* database have been extracted from images. Their nodes represent salient points in the images with their attributes being the features obtained by the point extractor. Edges have been deducted by Delaunay triangulation.

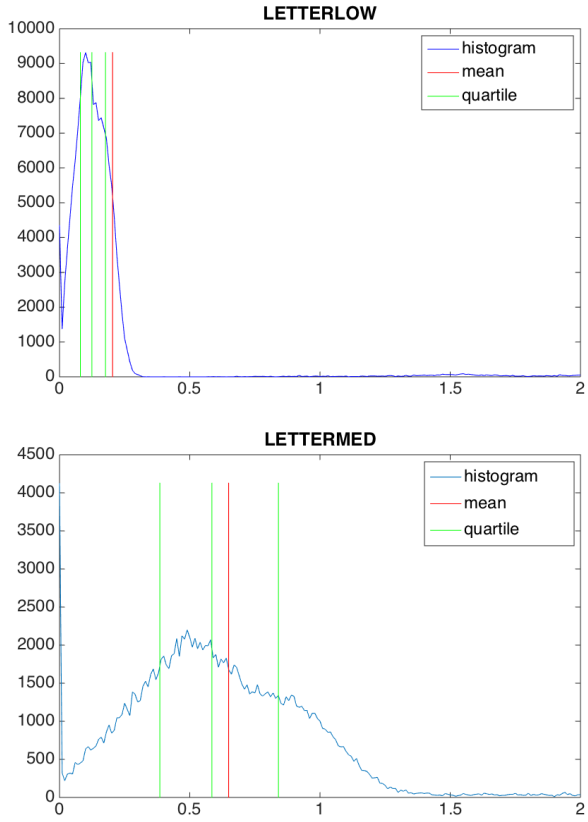
Table 2 shows the position of the quartiles, the mean and also half of the maximum values of the node substitution costs $d_n\left(v_a^p, v_i^q\right)$ given the whole correspondences. Clearly, if we want to hold equation 8, the insertion and deletion costs have to be defined such that $K_n \geq \frac{1}{2} Max$.

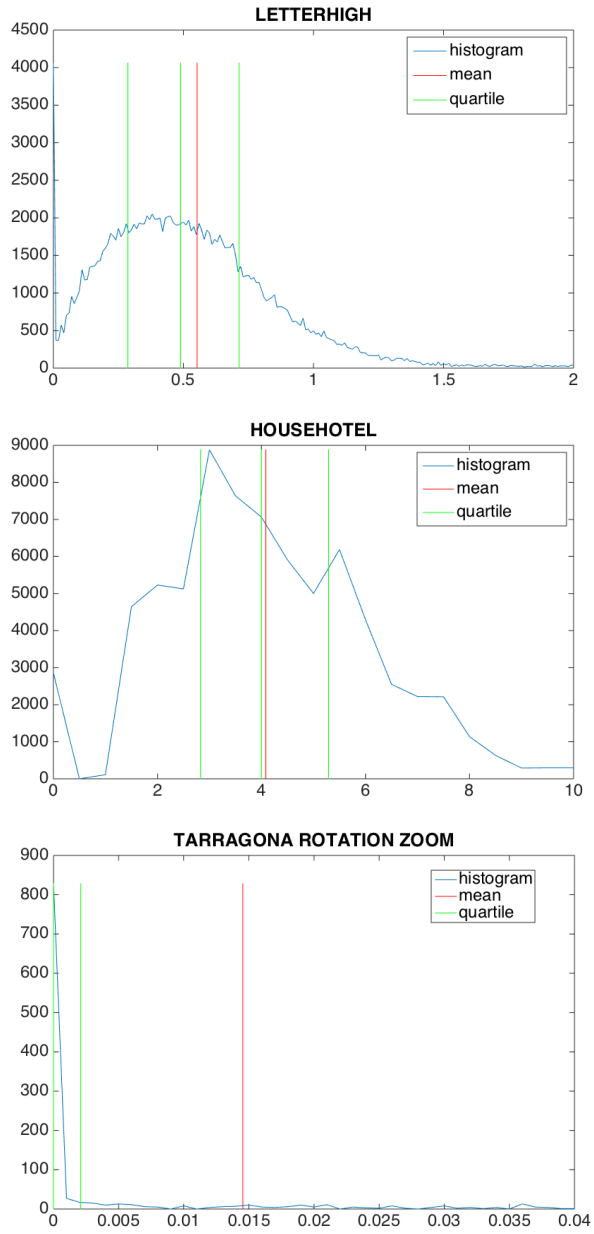
TABLE II. Node Substitution Costs

| | Q1 | Q2 | Q3 | Mean | ½ Max |
|---|---|---|---|---|---|
| **Letter low** | 0.08 | 0.12 | 0.17 | 0.20 | 1.68 |
| **Letter med** | 0.38 | 0.58 | 0.84 | 0.64 | 1.98 |
| **Letter high** | 0.28 | 0.48 | 0.71 | 0.55 | 1.98 |
| **House Hotel** | 2.82 | 4.00 | 5.29 | 4.08 | 5.75 |
| **Rotation Zoom** | 0 | 0 | 0.0021 | 0.0145 | 0.5 |

For the sake of clarification, Figure 1 shows the histogram of $d_n\left(v_a^p, v_i^q\right)$ given the whole databases with the quartiles and the mean values.

**Figure 1.** Histogram of node substitution costs in the five databases. In greed, we show the first three quartiles and in red the mean values.






We have used an error-tolerant graph-matching algorithm called Fast Bipartite [25] available in [28] to compute the optimal correspondence and the distance between the attributed graphs.

Table 3 shows the Hamming distance between the ground-truth correspondence and the automatically obtained correspondence when $K_n = Q1$, $K_n = Q2, K_n = Q3, K_n = Mean$, and $K_n = \frac{1}{2}Max$. Specific values are shown in table 2. The Hamming distance is computed as the number of node mappings that are different between both correspondences. Therefore, the lower these values, the better the performance.

We realise that the lowest Hamming distances are achieved in the positions of the insertion and deletion edit costs such that the triangle inequality is not hold, since these lowest Hamming distances are achieved in the first three quartiles, which are always smaller than $\frac{1}{2}Max$.

TABLE III. HAMMING DISTANCE

| | Q1 | Q2 | Q3 | Mean | ½ Max |
|---|---|---|---|---|---|
| **Letter low** | 0.6 | 0.6 | 0.6 | 0.6 | 0.7 |
| **Letter med** | 0.9 | 0.9 | 1.0 | 0.9 | 1.0 |
| **Letter high** | 0.9 | 0.8 | 0.9 | 0.9 | 1.2 |
| **House Hotel** | 0.61 | 0.71 | 0.78 | 0.72 | 0.80 |
| **Rotation Zoom** | 0.46 | 0.46 | 0.27 | 0.34 | 0.39 |

Table 4 shows the classification ratio using the same conditions than the previous experiments. To compute the classification ratio, we have used the reference and test set of each database and the 1-Nearest Neighbour classification algorithm. Recall that the *House Hotel* database does not have classes. It seems as the classification ratio performs similar to the Hamming distance. That is, the best values are achieved when the insertion and deletion edit costs are smaller than $\frac{1}{2}Max$.

The dependence between the recognition ratio and the Hamming distance between the ground truth and the obtained correspondences was explored in [20] while learning the edit costs. In that paper, it was empirically demonstrated that decreasing the Hamming distance leads the recognition ratio to increase. We have validated this dependence again. Moreover, the experimental validation in that paper shows that the optimisation method they presented converged to some negative insertion and deletion costs. Again, these values make the Graph Edit Distance not to be a truly defined distance.

TABLE IV. CLASSIFICATION RATIO

| | Q1 | Q2 | Q3 | Mean | ½ Max |
|---|---|---|---|---|---|
| **Letter low** | 0.97 | 0.97 | 0.97 | 0.97 | 0.93 |
| **Letter med** | 0.83 | 0.86 | 0.86 | 0.86 | 0.84 |
| **Letter high** | 0.74 | 0.82 | 0.83 | 0.82 | 0.74 |
| **House Hotel** | - | - | - | - | - |
| **Rotation Zoom** | 0.2 | 0.2 | 0.35 | 0.3 | 0.1 |

Finally, in table 5 we show the average runtime (in milliseconds) to compute one graph-to-graph comparison. We appreciate there is no relation, in general, between the insertion and deletion edit costs and the runtime.

TABLE V. AVERAGE RUNTIME TO MATCH A PAIR OF GRAPHS

| | Q1 | Q2 | Q3 | Mean | ½ Max |
|---|---|---|---|---|---|
| **Letter low** | 0.61 | 0.60 | 0.59 | 0.58 | 0.60 |
| **Letter med** | 0.63 | 0.59 | 0.60 | 0.59 | 0.63 |
| **Letter high** | 0.63 | 0.59 | 0.59 | 0.59 | 0.64 |
| **House Hotel** | 4.8 | 5.1 | 5.4 | 5.1 | 5.5 |
| **Rotation Zoom** | 15 | 15 | 10 | 8 | 7 |

## VII. Conclusions

Graph Edit Distance is nowadays the most widely used function to compare two graphs and to obtain a distance and a node correspondence. This function does not only depend on a pair of graphs, but also on the insertion and deletion edit costs on nodes and edges. These costs are usually defined as constants, and depending on their definition, we can consider the Graph Edit Distance is a true distance or not. The fact of not being a true distance can influence on the performance in some applications. Experimental validation has shown us that the insertion and deletion costs that obtains the lowest Hamming distances and the highest classification ratios are the ones where the triangle inequality is not hold and therefore, we conclude the Graph Edit Distance is not truly a distance. Therefore, some assumptions are not valid any more, for instance that $GED(G^p, G^q) \geq GED(G^p, G^t) + GED(G^t, G^q)$, which are commonly assumed on some applications such as graph retrieval.

## Acknowledgements

This research is supported by project DPI2013-42458-P and TIN2013-47245-C2-2-R, and by Consejo Nacional de Ciencia y Tecnología (CONACyT México).

## References

[1] A. Solé, F. Serratosa & A. Sanfeliu, On the Graph Edit Distance cost: Properties and Applications, International Journal of Pattern Recognition and Artificial Intelligence 26 (5), 1260004 [21 pages], 2012.

[2] Sanfeliu, A. and K.-S. Fu, A Distance measure between attributed relational graphs for pattern recognition. IEEE transactions on systems, man, and cybernetics, 1983. 13(3): p. 353-362.

[3] Donatello Conte, Pasquale Foggia, Carlo Sansone, Mario Vento: Thirty Years Of Graph Matching In Pattern Recognition. IJPRAI 18(3): 265-298 (2004).

[4] Mario Vento, "A long trip in the charming world of graphs for Pattern Recognition, Pattern Recognition", Available online 15 January 2014.

[5] P. Foggia, G. Percannella and M. Vento, Graph matching and learning in Pattern Recognition in the last 10 years, International Journal of Pattern Recognition and Artificial Intelligence, 2013.

[6] Gao, X., et al., A survey of graph edit distance. Pattern Analysis and applications, 2010. 13(1): p. 113-129.

[7] Bunke, H., Allermann ,G., "Inexact graph matching for structural pattern recognition", Pattern Recognition Letters, 1983. 1(4): p. 245–253.

[8] He, L., et al., Graph matching for object recognition and recovery. Pattern Recognition Letters, 2004. 37(7).

[9] F. Serratosa, X. Cortés & A. Solé, Component Retrieval based on a Database of Graphs for Hand-Written Electronic-Scheme Digitalisation, Expert Systems With Applications 40, pp: 2493 -2502, 2013.

[10] Berretti, S., Del Bimbo, A., Vicario, E.: Efficient Matching and Indexing of Graph Models in Content-Based Retrieval. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 10, pp. 1089-1105, 2001.

[11] Wong, A. and M. You, Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition. Transaction on Pattern Analysis and Machine Intelligence, 1985. PAMI-7(5): p. 599-609.

[12] Neuhaus, M. and H. Bunke, Automatic learning of cost functions for graph edit distance. Information Sciences, 2006. 177(1): p. 239-247.

[13] Serratosa, F., R. Alquézar, and A. Sanfeliu, Function-Described Graphs for modelling objects represented by attributed graphs. Pattern Recognition, 2003. 36(3): p. 781-798.

[14] Sanfeliu, A., F. Serratosa, and R. Alquézar, Second-Order Random Graphs for modelling sets of Attributed Graphs and their application to object learning and recognition. International Journal of Pattern Recognition and Artificial Intelligence, 2004. 18(3): p. 375-396.

[15] Bunke, H., On a relation between graph edit distance and maximum common subgraph. Pattern Recognition Letters, 1998. 18(8): p. 689-694.

[16] Bunke, H., Error Correcting Graph Matching: On the Influence of the Underlying Cost Function. Transactions on Pattern Analysis and Machine Intelligence, 1999. 21(9): p. 917-922.

[17] Caetano, T., et al., Learning Graph Matching. Transaction on Pattern Analysis and Machine Intelligence, 2009. 31(6): p. 1048-1058.

[18] M. Ferrer, F. Serratosa & K. Riesen, Improving Bipartite Graph Matching by Assessing the Assignment Confidence, Pattern Recognition Letters, 65, pp: 29-36, 2015.

[19] X. Cortés & F. Serratosa, Learning Graph Matching Substitution Weights based on the Ground Truth Node Correspondence, International Journal of Pattern Recognition and Artificial Intelligence, 30(2), pp: 1650005 [22 pages], 2016.

[20] X. Cortés & F. Serratosa, Learning Graph-Matching Edit-Costs based on the Optimality of the Oracle's Node Correspondences, Pattern Recognition Letters, 56, pp: 22 - 29, 2015.

[21] Jain, A.K. and D. Maltoni, Handbook of Fingerprint Recognition. 2003, Springer-Verlag New York.

[22] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals", Soviet Physics Doklady, Cybernetics and Control Theory, vol. 10, pp. 707-710, 1966.

[23] F. Serratosa, A. Sanfeliu, Signatures versus histograms: Definitions, distances and algorithms. Pattern Recognition 39 (5), pp: 921-934, 2006.

[24] Kaspar Riesen, Horst Bunke: Approximate graph edit distance computation by means of bipartite graph matching. Image Vision Comput. 27(7): 950-959 (2009).

[25] F. Serratosa, Fast Computation of Bipartite Graph Matching, Pattern Recognition Letters 45, pp: 244 - 250, 2014.

[26] Arkhangel'skii, A. V.; Pontryagin, L. S. (1990), General Topology I: Basic Concepts and Constructions Dimension Theory, Encyclopaedia of Mathematical Sciences, Springer, ISBN 3-540-18178-4.

[27] http://deim.urv.cat/~francesc.serratosa/databases/

[28] http://deim.urv.cat/~francesc.serratosa/SW