Optimal Transport vs Many-to-many assignment for Graph Matching

Anca-Ioana Grapa, Laure Blanc-Féraud, Ellen van Obberghen-Schilling, Xavier Descombes

Optimal Transport vs Many-to-many assignment for Graph Matching

Anca-Ioana GRAPA[1], Laure BLANC-FÉRAUD[2], Ellen VAN OBBERGHEN-SCHILLING[3], Xavier DESCOMBES[1]

[1]Université Côte d'Azur, Inria, CNRS, I3S, France

[2]Université Côte d'Azur, CNRS, Inria, I3S, France
Euclide B, 2000 Route des Lucioles, 06900 Sophia Antipolis, France

[3]Université Côte d'Azur, CNRS, INSERM, IBV, France
Parc Valrose, Faculté des Sciences, 06108 Nice Cedex 2, France

{agrapa,blancf,vanobber}@unice.fr, xavier.descombes@inria.fr

Abstract – Graph matching for shape comparison or network analysis is a challenging issue in machine learning and computer vision. Generally, this problem is formulated as an assignment task, where we seek the optimal matching between the vertices that minimizes the difference between the graphs. We compare a standard approach to perform graph matching, to a slightly-adapted version of regularized optimal transport, initially conceived to obtain the Gromov-Wassersein distance between structured objects (e.g. graphs) with probability masses associated to the nodes. We adapt the latter formulation to undirected and unlabeled graphs of different dimensions, by adding dummy vertices to cast the problem into an assignment framework. The experiments are performed on randomly generated graphs onto which different spatial transformations are applied. The results are compared with respect to the matching cost and execution time, showcasing the different limitations and/or advantages of using these techniques for the comparison of graph networks.

1 Introduction

With the ever-increasing use of graphs for representing complex data, due to their ability to synthetically capture structural relations, the need to provide a meaningful comparison among them becomes more relevant. In the computer science literature, this is often formulated as a graph-matching problem. The main idea is to obtain an evaluation of the similarity between two graphs, by finding the optimal correspondence between their vertices, such as to align their structure (i.e their adjacency matrices) [1]. In other words, the difference between them is evaluated by the amount of edge differences up to an optimal node permutation.

Among the methodologies proposed for this purpose, we consider the class of inexact graph matching, tailored specifically to real-world graph representations. This class of methods allows for a less strict correspondence of the graph vertices, and hence a less strict edge preservation. In this paper, we are interested in one of the various techniques to perform many-to-many graph matching [2], where the merging of multiple nodes to match another one is allowed, especially in the case of graphs with different dimensions (i.e. different number of total vertices). A particular case is considered to be one-to-one graph matching, where the issue of having to match graphs of different dimensions, is handled in practice by adding dummy nodes (with no connection) to the smaller size graph. In any case, these are formulated as discrete optimization problems for which several approximations are proposed [1].

From a different perspective, we refer to another line of work, mainly to approaches based on the optimal transport, for the comparison of structured objects (e.g. graphs) with associated probability distributions. The approach in [3] seeks to compare structured data, by minimizing the cost of transport of the mass from one discrete distribution to the other. For the comparison of distance/kernel matrices, they consider the intrinsic structural information in the cost formulation. Furthermore, an efficient algorithm to compute the Gromov-Wasserstein discrepancy [4] (an optimal transport metric capable of matching metric - measure spaces) is proposed.

In this paper, we compare the two approaches from a graph-matching perspective, on randomly generated graphs. If on one hand, the graph matching by many-to-many assignment captures the distance between two given graphs, in the cost of permuting the structure of one graph to match that of the second one (giving rise to an interpretable distance), on the other hand the Gromov-Wasserstein discrepancy is computed faster with similar performances in some of the scenarios. This work is also a preliminary study for the development of a graph-based computational model for the extracellular matrix fibronectin fibers in microscopy imaging [5].

2 Many-to-Many Graph Matching

We are motivated by the use of proven robust methods that can clearly reflect the cost of matching in their formulation, as a measure of similarity between the graphs. There are a few main approaches to find approximate solutions to the corresponding NP-hard problem, such as local search algorithms, spectral methods and optimization problems solved through a

continuous relaxation of the formulation [1]. In this paper, we consider the latter family of methods, for its advantages in terms of speed and, therefore, the possibility of comparing graphs with large dimensions in a reasonable time. Another reason to select these techniques, is their flexibility in terms of multiple matching of vertices (or addition of the dummy nodes with no connection), as opposed to the one-to-one matching framework. The problem is defined in the following manner. We consider two undirected graphs represented by their real-valued adjacency matrices G and H of size $N_G \times N_G$ and $N_H \times N_H$ respectively. Denoting by $||\cdot||_F$, the Frobenius norm of the matrices (defined as $||A||_F^2 = \text{tr}A^T A = (\sum_i \sum_j A_{ij}^2)$, the objective is to find the matrices $P_1 \in \{0,1\}^{N_K \times N_G}$ and $P_2 \in \{0,1\}^{N_K \times N_H}$ (which can be regarded as matching matrices between G, H, and a virtual intermediate graph for each matching, of size N_K, where N_K is min $\{N_G, N_H\}$):

$$\begin{aligned} &\min_{P1,P2} ||P_1 G^T P_1^T - P_2 H P_2^T||_F^2 \quad \text{s.t.} \\ &P_1 \mathbb{1}_{N_G} \leq k_{max} \mathbb{1}_{N_K}, \quad P_1^T \mathbb{1}_{N_K} = \mathbb{1}_{N_G} \\ &P_2 \mathbb{1}_{N_H} \leq k_{max} \mathbb{1}_{N_K}, \quad P_2^T \mathbb{1}_{N_K} = \mathbb{1}_{N_G} \end{aligned} \tag{1}$$

where $\mathbb{1}_N$ represents the constant N-dimensional vector of all-ones. In our experiments, we consider $N_G \geq N_H$. The maximal number of vertices merged together is represented by k_{max} and the many-to-many matching matrix is given by $P = P_1 P_2^T$, where $P \in \{0,1\}^{N_G \times N_H}$ is the matching matrix between G and H. The authors in [2] propose an approximation of the final solution, using a version of the conditional gradient algorithm, based on the continuous relaxation of (1). They also reformulate the gradient minimization as a linear assignment problem with a cubic complexity, hence making it feasible for high-dimension graph matching.

3 Gromov Wasserstein - Optimal Transport

3.1 General Optimal Transport

Optimal transport offers a well-founded approach to measure meaningful distances (expressed as transport costs) across discrete distributions. Its appeal comes from the ability to compare histograms while reflecting the geometry of the underlying space in the transport cost.

We define an N-dimensional set of histograms $S_N = \{x \in [0,1]^N, \sum_i x_i = 1\}$. Suppose that we take two discrete distributions a and $b \in S_N$, and we also consider the cost function matrix $C \in (\mathbb{R}^+)^{N \times N}$, a matrix whose term $C(i,j)$ denotes the cost to move the mass between bin i of a to the bin j of b. Then the classical formulation of the optimal transport between the two histograms a and b, that provides the well-known Earth Mover distance (EMD) [6], seeks the coupling (transportation) matrix P_{EMD} that satisfies: $\min_{P_{\text{EMD}} \in \mathbb{R}^{N \times N}} \sum_{ij} C_{ij} P_{\text{EMD}_{ij}}$, s.t. $P_{\text{EMD}} \mathbb{1}_N = a$, $P_{\text{EMD}}^T \mathbb{1}_N = b$ and $P_{\text{EMD}} \geq 0$. $P_{\text{EMD}_{ij}}$ represents the amount of mass moving from bin i of a to bin j of b. A generalization of this framework is the m-Wasserstein distance, defined as $W_m(a,b) = \min_{P_{\text{EMD}}} \langle P_{\text{EMD}}, C^m \rangle^{1/m}$. However, this classic discrete formulation doesn't take into account the inner structural dependency of the objects.

3.2 Structured Optimal Transport

Recently, within the framework of structured optimal transport, several works [3, 7, 8, 9] have shown the advantage of incorporating additional geometrical properties into the cost function, for tasks such as domain adaptation, natural language processing, computing graph barycenters or graph clustering. If the approach in [3] includes the intrinsic structure of the objects in the cost formulation, the authors in [7] present a new class of distances, that incorporates both structural and feature information into its transport cost. They focus on previously labeled structured objects, where for instance, graph edges represent relationships between features (vertices). Our interest in this paper is to compare unlabeled graphs, where no previous pairwise correspondences between the vertices of two graphs are known. Therefore, we focus on the work of Peyre et al.[3] that have considered a metric called Gromov-Wasserstein, capable of comparing objects that lie in spaces with different dimensions. This discrepancy is computed with a fast iterative algorithm based on an entropic regularization of the transportation matrix. In the following, we consider $(G,a) \in \mathbb{R}^{N_G \times N_G} \times S_{N_G}$ and $(H,b) \in \mathbb{R}^{N_H \times N_H} \times S_{N_H}$, where G and H encode the graphs' structure given by either the adjacency matrices, or the shortest path between the graph nodes, and a and b are the mass distributions associated to the graph vertices (e.g. uniform distributions, $a = \frac{1}{N_G} \mathbb{1}_{N_G}$ and $b = \frac{1}{N_H} \mathbb{1}_{N_H}$). The entropic Gromov-Wasserstein discrepancy between (G,a) and (H,b) is defined as follows:

$$\begin{aligned} &\min_{P'} \sum_{i,j,k,l} L(G_{i,k}, H_{j,l}) P'_{i,j} P'_{k,l} - \epsilon H(P') \quad \text{s.t.} \\ &P' \mathbb{1}_{N_H} = a, \quad P'^T \mathbb{1}_{N_G} = b \end{aligned} \tag{2}$$

where the entropy of the coupling matrix $P' \in \mathbb{R}^{N_G \times N_H}$ is: $H(P') = -\sum_{i,j} P'_{i,j} (\log(P'_{i,j}) - 1)$. The transport matrix indicates the matching between the two graphs such as if its term $P'(i,j) > 0$, the node j of graph H is assigned to the node i of graph G. The loss function $L(u,v)$ can be taken as the quadratic loss or Kullback-Leibler divergence. We note that for $\epsilon = 0$, the optimization problem is approximated by a classical solver with supercubical complexity.

4 Method design

In order to evaluate the performances of the two chosen methods (whose implementations are found online [1] [2]) in a graph

[1] http://projects.cbio.mines-paristech.fr/graphm/mtmgm.html
[2] https://github.com/gpeyre/2016-ICML-gromov-wasserstein

matching setting, we have first generated random graphs of different size (i.e. 16 vertices and 181 vertices). The graphs describe the structure of Voronoi diagrams, generated from seeds uniformly distributed on a bounded region. We applied several spatial transformations to the previously generated graphs (i.e rotation with $\pi/2$, removal of nodes) and evaluated the cost of matching between the original variants and the modified ones. The rotation is directly applied to the set of vertices, and therefore the size of the graphs remains unchanged. Subsequently, we chose to modify the size of the graphs by removing vertices of various degrees (the vertex degree is given by the number of incident edges). For each case, we have considered the two graphs to compare, G and H, (see Figures 1,2,3) represented alternatively by the binary adjacency matrix, or by the shortest path between the vertices at different orders: 2,3 and total. Additionally, we considered the integer values of the shortest path between nodes as well as the subunitary values (i.e. replacing the integer value by its inverse).

4.1 Graph Matching experiments

Denoting by F the objective function in (1) and by J_1, J_2 as all-ones matrices of size $N_K \times N_G$, and $N_K \times N_H$, the actual algorithm seeks the $min_{P1,P2}\{F-\lambda_s(||P_1-0.5J_1||_F^2+||P_2-0.5J_2||_F^2+c)\}$, where λ_s is a sparsity penalization parameter and c is depending on the graphs size. We tested the method for various values of λ_s in the $[0,1]$ interval, and chose the parameter configuration that resulted in the best matching. The difference between the graphs size is handled either by setting $k_{max} \geq 2$ (hence allowing at most k_{max} vertices to be merged), or by setting $k_{max} = 1$ (hence allowing the implicit choice of nodes that will be assigned as dummy, within the graph having a larger dimension). Setting $k_{max} \geq 2$ requires careful tuning of λ_s, and doesn't always guarantee a good matching quality, therefore, we kept $k_{max} = 1$. The initialization of the P_1 and P_2 matrices is extremely important as the non-convex problem is sensitive to it. In our experiments, we kept the initialization proposed by the authors, shown empirically to be a reasonable choice: $P_1 = \frac{1}{N_H}\mathbb{1}_{N_G}\mathbb{1}_{N_H}^T$ and P_2, the identity matrix I.

4.2 Gromov-Wasserstein experiments

For the loss function $L(u,v)$, we considered the quadratic version defined in [3], and uniform weights associated to the graph vertices (i.e. $a = \frac{1}{N_G}\mathbb{1}_{N_G}$ and $b = \frac{1}{N_H}\mathbb{1}_{N_H}$). We assume that a vertex i of graph G is matched to a vertex j of graph H, if its entire weight is transported to that of vertex j.

Regularizing the problem with an entropy term is shown to lead to a faster iterative algorithm. However, this also leads to a spread of mass from vertex i from graph G to multiple vertices of graph H. In order to recast this approach in the framework of graph-matching problems, we set $\epsilon = 0$. Moreover, to avoid the mass spreading as a consequence of the different dimensions of the two graphs (e.g when removing vertices), we added a dummy vertex (with no connection) to the graph with lower dimension (e.g. graph H) to which we assigned a mass that compensates the mass difference of vertices between the two graphs. Hence, this weight is equal to $1 - \frac{N_H}{N_G}$.

5 Results

For each of the previous pairs of graphs ($G - H$), we ran the algorithms for graph matching using many-to-many graph matching and structured optimal transport. In order to compare the performances of the two methods, we computed the cost of matching given by the difference between G, the adjacency matrix of the first graph, and PHP^T, the adjacency matrix of the matched graph, as $||G - PHP^T||_1$, where $||A||_1 = (\sum_i \sum_j |A_{ij}|)$.

FIG. 1: Initial graphs G - 16 and 181 nodes

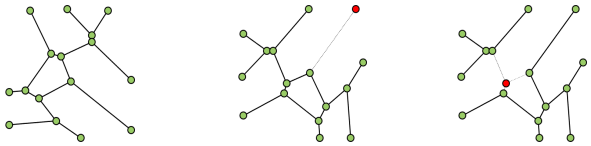

FIG. 2: Target graphs H - 16 nodes: Rotation with $\pi/2$ clockwise, Removal of one degree node, Removal of 3^{rd} degree node

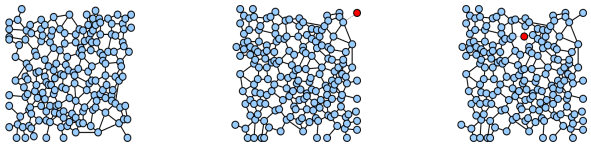

FIG. 3: Target graphs H - 181 nodes: Rotation with $\pi/2$ clockwise, Removal of one degree node, Removal of 4^{th} degree node

Since we already know which is the expected assignment between the vertices of the simulated graphs, we computed the cost of the perfect matching for all of the cases mentioned above. Table 1 contains the matching cost and execution time in the case of the perfect matching (PM), many-to-many method (MM), and the optimal transport (OT).

We notice in the case of many-to-many assignment, that for both graphs, the rotation is handled perfectly for most cases, except when G and H are binary adjacency matrices. Additionally, removal of one degree node returns the expected assignment, while removal of a higher degree node is correctly handled only in the case of larger graphs.

In the case of the approach based on the optimal transport, the results indicate that similarly to the first method, the rotation of the graphs is handled well, except for the case when

TAB. 1: Maching cost for the perfect matching (PM), many-to-many matching (MM) and optimal transport (OT). Graphs have either 16 or 181 vertices and the transformations are: rotation, removal of one-degree node (Remove A), removal of multiple-degree node (Remove B). G and H have the following representations: binary adjacency matrices (Int1), shortest path integer values and subunitary values at order 2,3, total (Int2, Int3, IntT, Sub2, Sub3, SubT).

Graph		PM	MM		OT	
Transf	**G,H**	**Cost**	**Cost**	**Time(ms)**	**Cost**	**Time(ms)**
Rotation (16)	**Int** 1	0	16	70	52	30
	Int 2	0	24	50	**0**	5
	Int 3	0	**0**	20	**0**	4
	IntT	0	**0**	20	**0**	2
	Sub 2	0	**0**	20	**0**	4
	Sub 3	0	**0**	20	**0**	3
	SubT	0	**0**	10	**0**	2
Remove A (16)	**Int** 1	2	**2**	20	58	70
	Int 2	10	**10**	20	86	7
	Int 3	34	**34**	20	186	6
	IntT	102	142	40	118	4
	Sub 2	4	**4**	30	32	5
	Sub 3	6.7	**6.7**	20	18	4
	SubT	10.4	**10.4**	20	17.5	2
Remove B (16)	**Int** 1	6	14	60	58	40
	Int 2	42	82	90	114	5
	Int 3	124	196	100	232	3
	IntT	538	546	100	550	2
	Sub 2	15	29	60	67	5
	Sub 3	23.6	39	70	44.3	4
	SubT	40.2	50	40	57.6	2
Rotation (181)	**Int** 1	0	424	5600	1048	100
	Int 2	0	**0**	1800	**0**	80
	Int 3	0	**0**	1500	**0**	30
	IntT	0	**0**	2700	**0**	20
	Sub 2	0	**0**	2000	384	100
	Sub 3	0	**0**	1500	14.7	50
	SubT	0	**0**	1700	**0**	20
Remove A (181)	**Int** 1	2	**2**	2300	1054	100
	Int 2	14	**14**	1900	4074	100
	Int 3	50	**50**	1500	**50**	40
	IntT	3766	**3766**	3000	5006	30
	Sub 2	5	**5**	5900	1000	100
	Sub 3	9	**9**	2000	29.7	80
	SubT	45.9	**45.9**	1900	105.2	40
Remove B (181)	**Int** 1	8	**8**	4800	1044	100
	Int 2	56	**56**	2800	4324	90
	Int 3	196	**196**	2000	**196**	70
	IntT	4928	**4928**	2900	5916	30
	Sub 2	20	**20**	18600	108	20
	Sub 3	34.7	**34.7**	3000	552	100
	SubT	94	**94**	2900	662	90

G and H are the adjacency matrices. A higher order of the shortest-path distance provides a more faithful representation of the graph adjacencies. However, in the case of larger-size graphs, increasing the order might also increase the time needed for the algorithm to converge to an optimal solution. This leads us to believe that a compromise (e.g. consider a 3^{rd} order) might work best, as confirmed by the results.

Generally, the optimal transport fails to provide the expected matching in far more scenarios than the many-to-many assignment. For most of the experiments nonetheless, the OT method finds the result in a shorter time, corresponding to one order of magnitude for the smaller graphs, and up to two orders of magnitude for the largest ones. This may be a considerable advantage over the many-to-many graph matching technique, if considered for the modeling of real graph-networks.

6 Conclusions

Generally, when it comes to comparing undirected and unlabeled graph networks in an assignment framework, the shortest path seems to be a better choice compared to the binary adjacency matrix (G - H representation), as it incorporates more information about their topological structure. In terms of the matching cost, many-to-many matching performs better, as highlighted by the results. However, we found that having to set a sparsity penalization parameter to trigger the expected solution as well as the sensitivity of the non-convex problem to the initialization, to be important drawbacks. The method by optimal transport performs significantly faster. Adding a dummy node to avoid mass splitting allows us to employ this formulation as a one-to-one graph matching problem. The results that we have obtained have a preliminary character, as we intend to explore these observations for further development of a computational model that is able to compare biological networks.

Acknowledgements. This work was supported by the French Government (National Research Agency, ANR) through the "Investments for the Future" LABEX SIGNALIFE: program reference ANR-11-LABX-0028-01.

References

[1] Junchi Yan, Xu-Cheng Yin, Weiyao Lin, Cheng Deng, Hongyuan Zha, and Xiaokang Yang. A short survey of recent advances in graph matching. pages 167–174, 06 2016.

[2] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. Many-to-many graph matching: A continuous relaxation approach. In *Machine Learning and Knowledge Discovery in Databases*, pages 515–530, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[3] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In *ICML 2016*, Proc. 33rd International Conference on Machine Learning, New-York, United States, June 2016.

[4] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, Aug 2011.

[5] Van Obberghen-Schilling E., Tucker R.P., Saupe F., Gasser I., Cseh B., and Orend G. Fibronectin and tenascin-c: Accomplices in vascular morphogenesis during development and tumor growth. *Int J Dev Biol*, 55:511–25, 2011.

[6] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000.

[7] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal Transport for structured data with application on graphs. *arXiv e-prints*, page arXiv:1805.09114, May 2018.

[8] David Alvarez-Melis, Tommi S. Jaakkola, and Stefanie Jegelka. Structured Optimal Transport. *arXiv e-prints*, page arXiv:1712.06199, December 2017.

[9] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. *arXiv e-prints*, page arXiv:1507.00504, Jul 2015.