Vol.2020-AVM-111 No.4
2020/11/26

最短経路長に基づく最適輸送問題を用いたグラフ類似度評価の検討

方 鐘熙[2,a)]　黄 健明[1,b)]　笠井 裕之[2,c)]

概要：ラベル付けされていないグラフ間の類似度測定の重要な課題の 1 つは，それぞれのグラフ内のノードの構造的同一性を捉えることである．グラフカーネルや Embedding などの既存のグラフマッチング手法の多くは，近傍の局所的な構造を扱いノードや辺のラベルや属性情報に依存しているため，性能が不十分である． そこで，ラベル無しグラフ上でのグラフ解析を目標とし，ノード間の最短経路長による単純な構造ノード特徴量表現手法を提案する．また，ノードの特徴量の類似性を容易に測定するために，SPL グラフ編集距離を提案する．次に，構造情報をサンプル分布に埋め込むことを目的として，最適輸送問題において，この SPL グラフ編集距離と ground distance を統合させた新しい OT ベースのグラフ類似度測定法を提案する．本稿では提案手法を SOT と呼ぶ．ラベルの無い実世界グラフデータセットを用いた評価実験において，提案方式 SOT は，最新の手法と比較して，ENZYMES と IMDB-M のデータセットにおいて，それぞれ 50%，36% の精度向上を達成した．また，ラベル付きグラフに対しても，他の手法と同等かそれ以上の性能を示すことを確認した．

OT-based Graph Similarity using Shortest-path Length

Abstract: One important challenge of similarity measure between unlabeled graphs is capturing the structural identity of nodes in the respective graphs. Most existing methods of graph matching such as graph kernel and embedding are adversely affected by unsatisfactory performance because they address the local structure of neighborhoods and rely on labels and attributes of nodes or edges. To this end, considering graph analysis tasks on unlabeled graphs, we propose a simple structural node representation by shortest-path length between nodes. We propose the SPL-graph edit distance to measure the node feature similarity easily. Then, aiming at embedding structural information into sample distributions, we propose a novel OT-based graph similarity measure by fusing this SPL-graph edit distance to the ground distance in optimal transport problem. As numerical evaluations reveal, the proposed method, designated as SOT, remarkably and stably outperforms state-of-the-art methods on several real-world unlabeled graphs, surprisingly gaining 50% and 36% accuracy improvement respectively in the ENZYMES and IMDB-M datasets. It also yields comparable or better performance than others for labeled graphs.

1. Introduction

Graph kernel [1], [2] and graph embedding [4], [5], [6], [7] have recently achieved remarkable progress in graph matching. However, many of them have two inherent limitations. The first limitation is that existing graph decomposition approaches and their simple similarity measures are likely to lose global topological information. In fact, many existing methods have been undertaken to compare neighborhood subgraphs centered at focused nodes, designated as *root nodes*. Consequently, they cannot efficiently handle cases in which a huge number of nodes are located far away from those root nodes. To alleviate this difficulty, some factorization-based methods attempt to seek low-dimensional representations of nodes assuming that the node adjacency matrix or connectivity matrix is *globally low-rank*. However, it is not always true when the matrix consists of a complex structure. A second limitation

[1] WASEDA University, School of Fundamental Science and Engineering, Dept. of Communications and Computer Engineering
[2] WASEDA University, Graduate School of Fundamental Science and Engineering
[a)] fzx@akane.waseda.jp
[b)] koukenmei@toki.waseda.jp
[c)] hiroyuki.kasai@waseda.jp

is that many of them rely heavily on labels and attributes of nodes or edges. However, many *unlabeled graphs* exist such as social networks, where individual nodes have no distinct identification or attributes, except through their interconnectivity.

In recent years, Optimal Transport (OT) is rapidly gaining popularity in a multitude of application areas in machine learning fields. One reason for this popularity is that many practical tasks in machine learning amount to calculation of the distance between probability measures or between their samples. This distance is called the *Wasserstein distance*. Another reason is that some recently developed approximate or regularized solvers have drastically reduced its potentially high computational cost. In a different line of issues, translation of graph-structured data into *structured* distributions is difficult to accomplish without loss of its structural information such as internal mutual relation among nodes. Although the Gromov–Wasserstein (GW) discrepancy [9] and its variant [10] represent graph-structured data exploiting its edge structure, their performances are not satisfactory, as the numerical section shows. The fused GW (FGW) [11] uses attributes and labels of nodes, but it also yields unsatisfactory results.

To this end, considering graph analysis tasks on unlabeled graphs, we propose a novel Wasserstein-distance-based graph similarity between two unlabeled graphs. More specifically, we first propose a simple structural node representation by *shortest-path lengths* between nodes, called SPL-node feature. Then, without embedding this SPL-node feature into a vector space, the *SPL-graph edit distance* is proposed for direct measurement of feature similarity between two nodes belonging to the two graphs. This distance is finally used as the ground distance to solve the optimal transport (OT) problem to derive the Wasserstein distance. The proposed method, designated as SOT, truly overcomes the shortcoming that the Wasserstein distance cannot capture such internal relations within its distribution.

Our contributions are summarized as explained below.

- We introduce a concept of shortest-path-based distance to extract structural features of a node to capture its connections to others in a local and global manner.
- We propose the SPL-graph edit distance and its simple calculation algorithm to measure similarity between two node features effectively by extending the notion of graph edit distance.
- We embed Graph-structured data into the same metric space to translate a graph matching problem into a discrete Monge's OT problem.

1.1 Related Work

The Wasserstein distance is an effective method for distribution comparison. A surge of approaches that are applicable to graph feature comparison has since occurred. However, it cannot directly capture the internal relation within the distribution. To address this difficulty, [9] proposed the Gromov–Wasserstein (GW) distance. However, the GW distance is non-convex. It is often NP-hard. [10] also proposed the GW discrepancy with an entropy-regularization term. It uses the projected gradient descent method using the Sinkhorn algorithm. The Fused Gromov–Wasserstein (FGW) [11] is an extension of the GW discrepancy to handle attributes and labels as well as structural information. GOT [12] was designed for graph alignment problems. It skillfully transforms the entirety of the graph into a normal distribution with zero mean value and Laplacian matrix variance. Also, it measures the distance between distributions according to the Wasserstein distance. The Wasserstein–Weisfeiler–Lehman Graph Kernels (WWL) [3] is the first method proposed to solve graph matching by transformation into the OT problem. This work demonstrated that the combination of the graph matching problem and the OT problem work together well. Although we followed the same line of this approach, our proposed similarity is measured by the simplified graph edit distance between the proposed structural node features in terms of local and global perspectives without reliance on labels and attributes. The resultant algorithm is calculable by linear complexity with a practical meaning.

2. Preliminaries

Graph $\mathcal{G}$ is structured data with a node set V and an edge set $E \subseteq V \times V$, denoted as $\mathcal{G}(V, E)$. The shortest path is the path between two nodes in the graph, of which the sum of the weights of the edges along its path is minimized. We refer to its length as the shortest-path length. The bold typeface lower-case and upper-case letters such as $\mathbf{x}$ and $\mathbf{X}$ respectively represent a vector and a matrix. Also, $\mathbf{X}_{i,j}$ represents the element at (i, j) of $\mathbf{X}$. $\mathbb{R}_+^n$ is the nonnegative n-dimensional vector, and $\mathbb{R}_+^{m\times n}$ denotes the nonnegative $m \times n$ size matrix. $\sum_n$ stands for the probability simplex with n bins. Also, δ_x is the Dirac function at x. $\langle \cdot and \cdot \rangle$ denote the Euclidean dot-product between

vectors. $\circ$ is the Hadamard product. $||\cdot||_2$ is ℓ_2 norm.

Definition 1 (Graph edit distance [8]). *Graph edit distance (GED) is a measure of the graph dissimilarity. The traditional GED is defined as the sum of the minimum operation cost of deleting, inserting, and substituting nodes and edges necessary to transform one graph into another.*

Definition 2 (Optimal Transport). *Optimal transportation (OT) is derived from the Monge's OT problem, which is intended to obtain the minimum transportation cost and transport plan between two distributions. Because of the strict conditions of Monge's OT problem, it is difficult to solve. The existing OT usually refers to Kantorovich's OT problem.*

We define two simplexes of histograms with m and n on the same matrix space. They are defined as $\sum_m = \{\mathbf{a} \in \mathbb{R}_+^m : \sum_{i=1}^m \mathbf{a}_i = 1\}$ and $\sum_n = \{\mathbf{b} \in \mathbb{R}_+^n : \sum_{j=1}^n \mathbf{b}_j = 1\}$. Then we define two probability measures

$$\alpha = \sum_{i=1}^{m} \mathbf{a}_i \delta_{x_i} \quad \text{and} \quad \beta = \sum_{j=1}^{n} \mathbf{b}_j \delta_{x_j},$$

where $x_i \neq x_j$ for $i \neq j$ is assumed without loss of generality. We denote the ground cost matrix as $\mathbf{C} \in \mathbb{R}_+^{m \times n}$. The element at (i, j) of the $\mathbf{C}$ represents the transport cost between bin i and bin j (locations x_i and y_j), where the $\mathbf{C}_{i,j}$ is also called the ground distance. The optimal transport between two histograms is defined as

$$L_{\mathbf{C}}(\mathbf{a}, \mathbf{b}) = \min_{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j}, \qquad (1)$$

where $\mathbf{U}(\mathbf{a}, \mathbf{b})$ is defined as

$$\mathbf{U}(\mathbf{a}, \mathbf{b}) = \{\mathbf{P} \in \mathbb{R}^{|V_A| \times |V_B|} : \mathbf{P} \mathbb{1}_{|V_A|} = \mathbf{a},\ \mathbf{P}^{\mathrm{T}} \mathbb{1}_{|V_B|} = \mathbf{b}\},$$

where $\mathbf{P}$ is a coupling matrix that describes the transport plan. Eq. (1) is a linear programming problem and a convex optimization problem. We can add the entropy term $\mathbf{H}(\mathbf{P}) = -\sum_{i,j} \mathbf{P}_{i,j}(\log(\mathbf{P}_{i,j} - 1))$ to Eq. (1), then Eq. (1) can be reformulated as

$$L_{\mathbf{C}}^{\varepsilon}(\mathbf{a}, \mathbf{b}) = \min_{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j} - \varepsilon \mathbf{H}(\mathbf{P}), \qquad (2)$$

where $\varepsilon (> 0)$ is the learning rate. When ε is close to 0, the solution of Eq. (2) is close to the original solution of Eq. (1). This problem is solvable using the Sinkhorn algorithm [13], [14]. Finally, the solution $\mathbf{P}^*$ is the optimal transport plan. The total transport cost of $\langle \mathbf{C}, \mathbf{P}^* \rangle$ is equal to the Wasserstein distance $\mathcal{W}(\alpha, \beta)$.

3. Shortest-path Length Node Feature and Its Graph Edit Distance

This section presents a proposal of the shortest-path length node feature, which represents internal structural relations of a node with other nodes within a graph. Then, by extending the notion of the graph edit distance defined, the SPL-graph edit distance to measure the dissimilarity between two shortest-path length node features is proposed. This obtained distance is used to form the ground distance $\mathbf{C}$ for the proposed Wasserstein graph similarity in the succeeding section.

3.1 Shortest-path length node feature

When considering a distance between two graphs, a common method is measurement of the distance between features of the corresponding two nodes belonging to these two graphs. This paper is a particular attempt to seek a *relative position* or *relative structural identity* of a node inside the graph, and to *encode* it by a simple but powerful descriptor. For this particular purpose, we first introduce a new concept of *shortest-path length node feature*, denoted as SPL-node feature in short, which consists of shortest-path lengths of one node with all remaining nodes within the graph.

More concretely, we calculate all the shortest-path lengths from the focused node to all the remaining nodes, and encode the set of these lengths as the descriptor. The shortest-path length between the i-th node and the j-th node, where $j \neq i$ and $j \in |V|$ is designated as $Sp(v_i, v_j) \in \mathbb{Z}$, where $v_i, v_j \in V$, which refers to the i-th node in the graph. Consequently, the SPL-node feature $\mathcal{F}$ of node v_i in the graph is defined as

$$\mathcal{F}(v_i) = \{Sp(v_i, v_j) \mid j \in |V|, j \neq i\} \quad \text{for } i = 1, \ldots, |V|.$$

3.2 SPL-graph edit distance for node similarity

This subsection presents derivation of node similarity between two SPL-node features. As the formulation of $\mathcal{F}(v_i)$ shows, the SPL-node features of two nodes have different lengths. Vector norms cannot be applied directly to measure the feature distances. Moreover, even if the two features have the same length, simple vector comparison does not reflect the meanings and the structure of the SPL-node feature. Therefore, we decided to introduce the structural concept into the SPL-node features that are derived originally from $\mathcal{F}(v_i)$. We therefore regard the

SPL-node feature as a *graph*[*1]. Therefore, one possible and effective approach to measure the feature distance is the conventional GED, which usually refers to the minimum operation cost to convert one graph to another, or vice versa. Here, denoting this distance between graphs $\mathcal{G}_A(V_A, E_A)$ and $\mathcal{G}_B(V_B, E_B)$ as $d_{\mathcal{G}_A,\mathcal{G}_B}$, the graph edit distance can be formally defined as

$$d_{\mathcal{G}_A,\mathcal{G}_B} = \min_{(e_1,\ldots,e_Q)\in\mathcal{P}(\mathcal{G}_A,\mathcal{G}_B)} \sum_{q=1}^{Q} c(e_q),$$

where $\mathcal{P}(\mathcal{G}_A, \mathcal{G}_B)$ denotes the set of edit operation paths to convert $\mathcal{G}_A$ to $\mathcal{G}_B$. Also, e_q stands for the q-th edit operation along the edit path, where the operation includes deletion, insertion, and substitution of nodes or edges. In addition, Q is the total number of the edit operations; $c(e_q)$ represents the cost of the edit operation e_q. Because this is known to be NP-hard, several methods have been proposed to obtain an approximate solution. Among them, a possible approach is *string edit distance* e.g., Levenshtein-distance [15], against sorted $\mathcal{F}$, which is also an approximate solution of GED. However, in general, we face high computational costs attributable to the application of sophisticated algorithms, e.g., dynamic programming.

Consequently, we propose a *SPL-graph edit distance*, denoted as d^{SPL}, by extending the notion of GED. It is efficient to represent the similarity between two nodes, which will be revealed in the numerical section. For this SPL-graph edit distance, we first introduce the notion of *SPL-graph* $\mathcal{S}_i$ of which the root node is the focused node v_i. The main difference from the original $\mathcal{G}$ is that the remaining other nodes $v_j (i \neq j)$ connected directly to v_i with a single edge with length equal to the shortest-path length between them in the original $\mathcal{G}$.

We propose an algorithm to calculate the SPL-graph edit distance between SPL-graphs $\mathcal{S}_A$ and $\mathcal{S}_B$, i.e., $d^{\mathrm{SPL}}_{\mathcal{S}_A,\mathcal{S}_B}$. Noteworthy points in the design of the algorithm are summarized.

- **Disable substitution operation:** We disable the substitution operation. Only insertion and deletion operations are allowed. This simple modification in the operational strategy gives us an extremely simple calculation algorithm of the distance, and enables significant reductions of computational costs by avoiding complicated algorithms such as dynamic programming.
- **Simple distance calculation:** We execute insertion and deletion operations *only* when the number of nodes with equal lengths to the root node are different in the two SPL-graphs. Consequently, these different numbers, the different frequencies of length values in $\mathcal{F}(v_i)$, are fused directly into the distance calculation (refer to (a) in Eq. (3)).
- **Weighting operational cost:** We must also incorporate consideration of the amplitude of the value in $\mathcal{F}(v_i)$ on which the edit operation is performed. The edge length in the SPL-graph represents the structural distance from the focused node with others. Therefore, the operation costs on different lengths must be differentiated, such as the costs between those of the edge length "1" and "10" (refer to (b) in Eq. (3)).
- **Symmetric operational cost:** We guarantee the symmetric property of the costs for the conversion between two graphs. Therefore, we set $c(e_q) = C(\in \mathbb{R}_+)$ for all q so that the costs of the insertion and deletion operations are the same (refer to (c) in Eq. (3)).

Consequently, denoting the number of the element of which value is equal to k as $\phi(k)$, $d^{\mathrm{SPL}}_{\mathcal{S}_i,\mathcal{S}_j}$ is calculated as

$$\begin{aligned} d^{\mathrm{SPL}}_{\mathcal{S}_i,\mathcal{S}_j} &:= d^{\mathrm{SPL}}(\mathcal{F}(v_i), \mathcal{F}(v_j)) \\ &:= \underbrace{C}_{(c)} \sum_{k=1}^{K} \underbrace{k}_{(b)} \cdot \underbrace{|\phi_i(k) - \phi_j(k)|}_{(a)}, \end{aligned} \tag{3}$$

where $\phi_i(\cdot)$ and $\phi_j(\cdot)$ respectively denote the values of $\phi(\cdot)$ corresponding to $\mathcal{S}_i$ and $\mathcal{S}_j$. Furthermore, K is defined as

$$K = \max(n_{i\max}, n_{j\max}),$$

where $n_{i\max}$ and $n_{j\max}$ respectively denote the largest number in $\mathcal{F}(\mathcal{S}_A)$ and $\mathcal{F}(\mathcal{S}_B)$.

4. Wasserstein Graph Similarity

This section presents the Wasserstein graph similarity exploiting the SPL-graph edit distance as the ground cost matrix $\mathbf{C}$ in the optimal transport problem. The proposed similarity enables us to perform various graph analysis tasks such as graph matching and graph alignment.

4.1 Optimal Transport Matrix for graph similarity

The proposed Wasserstein graph similarity measures

[*1] For this explanation, we adopt the term "SPL-graph" to represent graph edit distance. In fact, as the practical algorithm to calculate the distance shows later, it does *not necessarily* require this notion. The calculation can perform directly on $F(v_i)$. However, for ease of explanation of the proposed algorithm, we adhere to use the notion of "graph" by following the operation of GED.

the distances to match the nodes of one graph with the nodes of another, which is formulated as the graph node alignment problem. This is analogous to the discrete Monge's OT problem.

Presuming that we have two graphs $\mathcal{G}_A(V_A, E_A)$ and $\mathcal{G}_B(V_B, E_B)$. We embed $\mathcal{G}_A$ and $\mathcal{G}_B$ into the same metric space. All the nodes of the two graphs are located respectively at points in $\{x_1, \ldots, x_{|V_A|}\} \in \mathcal{X}$ and $\{y_1, \ldots, y_{|V_B|}\} \in \mathcal{Y}$. Here, we consider each node of $\mathcal{G}_A$ as the starting point. We also consider each node of $\mathcal{G}_B$ as the endpoint for the transport setting in the optimal transport problem. The two histograms of $\mathbf{a}$ and $\mathbf{b}$ are defined respectively in the probability simplex $\sum_{|V_A|}$ and $\sum_{|V_B|}$. Furthermore, as a discrete measure α with weights $\mathbf{a}$ on the locations $x_i, \ldots, x_{|V_A|}$, we write $\alpha = \sum_{i=1}^{|V_A|} \mathbf{a}_i \delta_{x_i}$, where α represents the distribution of nodes in $\mathcal{G}_A$, and where $\mathbf{a}_i$ refers to the relative importance of the node. Similarly, we define the measure $\beta = \sum_{j=1}^{|V_B|} \mathbf{b}_j \delta_{y_j}$.

Finally, inputting the SPL-graph edit distance into $\mathbf{C} \in \mathbb{R}_+^{|V_A| \times |V_B|}$ in Eq. (2), one obtains $\mathbf{P}^*$ by calculating

$$\mathbf{P}^* = \underset{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})}{\arg\min} \sum_{i \in [|V_A|], j \in [|V_B|]} d^{\mathrm{SPL}}(\mathcal{F}_A(v_i), \mathcal{F}_B(v_j)) \cdot \mathbf{P}_{i,j}. \tag{4}$$

4.2 Application to graph matching

We introduce two approaches of the graph similarity measure and their classification method. One is a common method that uses the Wasserstein-distance directly to measure two SPL-distributions is given as

$$\mathcal{W}(\mathcal{G}_A, \mathcal{G}_B) = \langle \mathbf{C}, \mathbf{P}^* \rangle. \tag{5}$$

We combine Eq. (5) with the Laplacian kernel function to construct SOT kernel (SOT-K), defined as

$$K_{\mathrm{SOT}}(\mathcal{G}_A, \mathcal{G}_B) = e^{-\lambda \mathcal{W}(\mathcal{G}_A, \mathcal{G}_B)}.$$

We also proposed another approach to perform classification using RBF kernel for SVM, which receives the graph distances as a feature vector. In this case, this operation also makes the ratio of between-class distance to within-class distance larger because ℓ_2 norm enlarges the ratio of large distance to small distance. This operation might perform better under the RBF kernel. Therefore, the new graph distance is given as

$$\mathcal{L}(\mathcal{G}_A, \mathcal{G}_B) = ||\mathbf{C} \circ \mathbf{P}^*||_2. \tag{6}$$

We use Eq. (6) to construct a graph distance feature vector, defined as

$$E_{\mathrm{SOT}}(\mathcal{G}_i) = \{\mathcal{L}(\mathcal{G}_i, \mathcal{G}_j)\}_{j=1}^{N}.$$

We designate this graph embedding method as SOT embedding (SOT-E).

参考文献

[1] Haussler, D.: Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California, (1999).
[2] Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K. and Borgwardt, K. M.: Weisfeiler-lehman graph kernels. Journal of Machine Learning Research, Vol. 12, No. 9, 2011.
[3] Togninalli, M., Ghisu, E., Llinares-López, F., Rieck, B. and Borgwardt, K.: Wasserstein Weisfeiler–Lehman Graph Kernels, Advances in Neural Information Processing Systems (NeurIPS), 2019.
[4] Perozzi, B., Al-Rfou, R. and Skiena, S.: Deepwalk: Online learning of social representations, Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp. 701–710, 2014.
[5] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. and Mei, Q.: Line: Large-scale information network embedding, Proceedings of the 24th international conference on world wide web (WWW), 2015.
[6] Grover, A. and Leskovec, J.: node2vec: Scalable feature learning for networks, Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), 2016.
[7] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient estimation of word representations in vector space arXiv preprint arXiv:1301.3781, 2013.
[8] Sanfeliu, A. and Fu, K.-S.: A distance measure between attributed relational graphs for pattern recognition, IEEE transactions on systems, man, and cybernetics, No. 3, pp. 353–362 (1983).
[9] Mémoli, Facundo: Gromov–Wasserstein distances and the metric approach to object matching, Foundations of computational mathematics, No. 4, pp. 417–487, Springer, Vol. 11, 2011.
[10] Peyré, G., Cuturi, M. and Solomon, J.: Gromov-wasserstein averaging of kernel and distance matrices, International Conference on Machine Learning (ICML), 2016.
[11] Vayer, T., Chapel, L., Flamary, R., Tavenard, R. and Courty, N.: Fused Gromov-Wasserstein distance for structured objects: theoretical foundations and mathematical properties, arXiv preprint arXiv:1811.02834, 2018.
[12] Maretic, H. P., Gheche, M. E., Chierchia, G. and Frossard, P.: GOT: an optimal transport framework for graph comparison, Advances in Neural Information Processing Systems (NeurIPS), 2019.
[13] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport, Advances in neural information processing systems (NIPS), 2013.
[14] Sinkhorn, R. and Knopp, P.: Pacific Journal of Mathematics, No. 2, pp. 343–348, Concerning nonnegative matrices and doubly stochastic matrices, Mathematical Sciences Publishers, Vol. 21, (1967).
[15] Levenshtein, V. I.: Soviet physics doklady, No. 8, pp. 707–710, Binary codes capable of correcting deletions, insertions, and reversals, Vol. 10, (1966).