Gromov-Wasserstein Learning for Graph Matching and Node Embedding

Hongteng Xu [1 2] **Dixin Luo** [2] **Hongyuan Zha** [3] **Lawrence Carin** [2]

Abstract

A novel Gromov-Wasserstein learning framework is proposed to jointly match (align) graphs and learn embedding vectors for the associated graph nodes. Using Gromov-Wasserstein discrepancy, we measure the dissimilarity between two graphs and find their correspondence, according to the learned optimal transport. The node embeddings associated with the two graphs are learned under the guidance of the optimal transport, the distance of which not only reflects the topological structure of each graph but also yields the correspondence across the graphs. These two learning steps are mutually-beneficial, and are unified here by minimizing the Gromov-Wasserstein discrepancy with structural regularizers. This framework leads to an optimization problem that is solved by a proximal point method. We apply the proposed method to matching problems in real-world networks, and demonstrate its superior performance compared to alternative approaches.

1. Introduction

Real-world entities and their interactions are often represented as graphs. Given two or more graphs created in different domains, graph matching aims to find a correspondence across different graphs. This task is important for many applications, $e.g.$, matching the protein networks from different species (Sharan & Ideker, 2006; Singh et al., 2008), linking accounts in different social networks (Zhang & Philip, 2015), and feature matching in computer vision (Cordella et al., 2004). However, because it is NP-hard, graph matching is challenging and often solved heuristically. Further complicating matters, the observed graphs may be noisy ($e.g.$, containing unreliable edges), which leads to unsatisfying matching results using traditional methods.

[1]Infinia ML, Inc., Durham, NC, USA [2]Department of ECE, Duke University, Durham, NC, USA [3]College of Computing, Georgia Institute of Technology, Atlanta, GA, USA. Correspondence to: Hongteng Xu <hongtengxu313@gmail.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

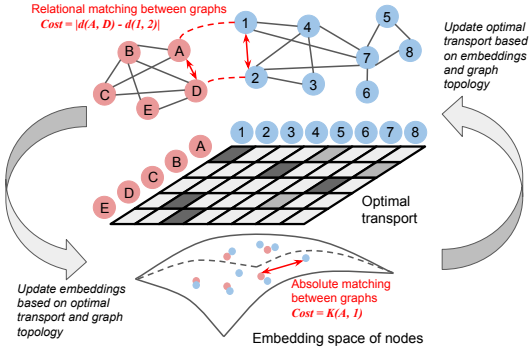


Figure 1. An illustration of the proposed method.

A problem related to graph matching is the learning of node embeddings, which aims to learn a latent vector for each graph node; the collection of embeddings approximates the topology of the graph, with similar/related nodes nearby in embedding space. Learning suitable node embeddings is beneficial for graph matching, as one may seek to align two or more graphs according to the metric structure associated with their node embeddings. Although graph matching and node embedding are highly related tasks, in practice they are often treated and solved independently. Existing node embedding methods (Perozzi et al., 2014; Tang et al., 2015; Grover & Leskovec, 2016) are designed for a single graph, and applying such methods separately to multiple graphs doesn't share information across the graphs, and, hence, is less helpful for graph matching. Most graph matching methods rely purely on topological information ($i.e.$, adjacency matrices of graphs) and ignore the potential functionality of node embeddings (Kuchaiev et al., 2010; Neyshabur et al., 2013; Nassar et al., 2018). Although some methods consider first deriving embeddings for each graph and then learning a transformation between the embeddings, their results are often unsatisfying because their embeddings are predefined and the transformations are limited to orthogonal projections (Grave et al., 2018) or rigid/non-rigid deformations (Myronenko & Song, 2010).

This paper considers the joint goal of graph matching and learning node embeddings, seeking to achieve improvements in both tasks. As illustrated in Figure 1, to achieve this goal we propose a novel Gromov-Wasserstein learning framework. The dissimilarity between two graphs is measured by the Gromov-Wasserstein discrepancy (GW discrepancy) (Peyré et al., 2016), which compares the distance ma-

trices of different graphs in a relational manner, and learns an optimal transport between the nodes of different graphs. The learned optimal transport indicates the correspondence between the graphs. The embeddings of the nodes from different graphs are learned jointly: the distance between the embeddings within the same graph should approach the distance matrix derived from data, and the distance between the embeddings across different graphs should reflect the correspondence indicated by the learned optimal transport. As a result, the objectives of graph matching and node embedding are unified as minimizing the Gromov-Wasserstein discrepancy between two graphs, with structural regularizers. This framework leads to an optimization problem that is solved via an iterative process. In each iteration, the embeddings are used to estimate distance matrices when learning the optimal transport, and the learned optimal transport regularizes the learning of embeddings in the next iteration.

There are two important benefits to tackling graph matching and node embedding jointly. First, the observed graphs often contain spurious edges or miss some useful edges, leading to noisy adjacency matrices and unreliable graph matching results. Treating the distance between learned node embeddings as complementary information of observed edges, we can approximate the topology of graph more robustly, and accordingly, match noisy graphs. Second, our method regularizes the GW discrepancy and learns embeddings of different graphs on the same manifold, instead of learning an explicit transformation between the embeddings with predefined constraints. Therefore, the proposed method is more flexible and has lower risk of model misspecification ($i.e.$, imposing incorrect constraints on the transformation); the distance between the embeddings of different graphs can be calculated directly without any additional transformation. We test our method on real-world matching problems and analyze its performance in depth. Experiments show that our method obtains encouraging matching results, with comparisons made to alternative approaches.

2. Gromov-Wasserstein Learning Framework

Assume we have two sets of entities (nodes), denoted as source set $\mathcal{V}_s$ and target set $\mathcal{V}_t$. Without loss of generality, we assume that $|\mathcal{V}_s| \leq |\mathcal{V}_t|$. For each set, we observe a set of interactions between its entities, $i.e.$, $\mathcal{E}_k = \{(v_i, v_j, w_{ij}) | v_i, v_j \in \mathcal{V}_k\}$, where $k = s$ or t, and w_{ij} counts the appearances of the interaction (v_i, v_j). Accordingly, the data of these entities can be represented as two graphs, denoted as $G(\mathcal{V}_s, \mathcal{E}_s)$ and $G(\mathcal{V}_t, \mathcal{E}_t)$, and we focus on the following two tasks: i) Find a correspondence between the graphs. ii) Obtain node embeddings of the two graphs, $i.e.$, $\boldsymbol{X}_s = [\boldsymbol{x}_i^s] \in \mathbb{R}^{D \times |\mathcal{V}_s|}$ and $\boldsymbol{X}_t = [\boldsymbol{x}_i^t] \in \mathbb{R}^{D \times |\mathcal{V}_t|}$. These two tasks are unified in a framework based on Gromov-Wasserstein discrepancy.

2.1. Gromov-Wasserstein discrepancy

Gromov-Wasserstein discrepancy was proposed in (Peyré et al., 2016), which is a natural extension of Gromov-Wasserstein distance (Mémoli, 2011). Specifically, the definition of Gromov-Wasserstein distance is as follows:

Definition 2.1. *Let (X, d_X, μ_X) and (Y, d_Y, μ_Y) be two metric measure spaces, where (X, d_X) is a compact metric space and μ_X is a Borel probability measure on X (with (Y, d_Y, μ_Y) defined in the same way). The Gromov-Wasserstein distance $d_{GW}(\mu_X, \mu_Y)$ is defined as*

$$\inf_{\pi \in \Pi(\mu_X, \mu_Y)} \iint_{X \times Y, X \times Y} L(x, y, x', y') \, d\pi(x, y) \, d\pi(x', y'),$$

where $L(x, y, x', y') = |d_X(x, x') - d_Y(y, y')|$ is the loss function and $\Pi(\mu_X, \mu_Y)$ is the set of all probability measures on $X \times Y$ with μ_X and μ_Y as marginals.

This defines an optimal transport-like distance (Villani, 2008) for metric spaces: it calculates distances between pairs of samples within each domain and measures how these distances compare to those in the other domain. It does not require one to directly compare the samples across different spaces and the target spaces can have different dimensions. When d_X and d_Y are replaced with dissimilarity measurements rather than strict distance metrics, and the loss function L is defined more flexibly, $e.g.$, mean-square-error (MSE) or KL-divergence, we relax the Gromov-Wasserstein distance to the proposed Gromov-Wasserstein *discrepancy*. These relaxations make the proposed Gromov-Wasserstein learning framework suitable for a wide range of machine learning tasks, including graph matching.

In graph matching, a metric-measure space corresponds to the pair $(\boldsymbol{C}, \boldsymbol{\mu}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|} \times \Sigma^{|\mathcal{V}|}$ of a graph $G(\mathcal{V}, \mathcal{E})$, where $\boldsymbol{C} = [c_{ij}] \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ represents a distance/dissimilarity matrix derived according to the interaction set $\mathcal{E}$, $i.e.$, each c_{ij} is a function of w_{ij}. The empirical distribution of nodes, $i.e.$, $\boldsymbol{\mu} = [\mu_i] \in \Sigma^{|\mathcal{V}|}$, is calculated based on the normalized degree of graph. It reflects the probability of each node appearing in observed interactions. Given two graphs $G(\mathcal{V}_s, \mathcal{E}_s)$ and $G(\mathcal{V}_t, \mathcal{E}_t)$, the Gromov-Wasserstein discrepancy between $(\boldsymbol{C}_s, \boldsymbol{\mu}_s)$ and $(\boldsymbol{C}_t, \boldsymbol{\mu}_t)$ is defined as

$$\begin{aligned} & d_{GW}(\boldsymbol{\mu_s}, \boldsymbol{\mu_t}) \\ & := \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu_s}, \boldsymbol{\mu_t})} \sum\nolimits_{i,j,i',j'} L(c_{ij}^s, c_{i'j'}^t) T_{ii'} T_{jj'} \qquad (1) \\ & = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu_s}, \boldsymbol{\mu_t})} \langle \boldsymbol{L}(\boldsymbol{C}_s, \boldsymbol{C}_t, \boldsymbol{T}), \boldsymbol{T} \rangle. \end{aligned}$$

Here, $\Pi(\boldsymbol{\mu}_s, \boldsymbol{\mu}_t) = \{\boldsymbol{T} \in \mathbb{R}^{|\mathcal{V}_s| \times |\mathcal{V}_t|} \mid \boldsymbol{T}\boldsymbol{1}_{|\mathcal{V}_t|} = \boldsymbol{\mu}_s, \boldsymbol{T}^\top \boldsymbol{1}_{|\mathcal{V}_s|} = \boldsymbol{\mu}_t\}$. $L(\cdot, \cdot)$ is an element-wise loss function, with typical choices the square loss $L(a, b) = (a - b)^2$ and the KL-divergence $L(a, b) = a \log \frac{a}{b} - a + b$. Accordingly, $\boldsymbol{L}(\boldsymbol{C}_s, \boldsymbol{C}_t, \boldsymbol{T}) = [L_{jj'}] \in \mathbb{R}^{|\mathcal{V}_s| \times |\mathcal{V}_t|}$ and each

$L_{jj'} = \sum_{i,i'} L(c^s_{ij}, c^t_{i'j'})T_{ii'}$, and $\langle \cdot, \cdot \rangle$ represents the inner product of matrices; $\boldsymbol{T}$ is the optimal transport between the nodes of two graphs, and its element T_{ij} represents the probability that $v_i \in \mathcal{V}_s$ matches $v_j \in \mathcal{V}_t$. By choosing the largest T_{ij} for each i, we find the correspondence that minimizes the GW discrepancy between the two graphs.

However, such a graph matching strategy raises several issues. First, for each graph, its observed interaction set can be noisy, which leads to an unreliable distance matrix. Minimizing the GW discrepancy based on such distance matrices has a negative influence on matching results. Second, the Gromov-Wasserstein discrepancy compares different graphs relationally based on their edges ($i.e.$, the distance between a pair of nodes within each graph), while most existing graph matching methods consider the information of nodes *and* edges jointly (Neyshabur et al., 2013; Vijayan et al., 2015; Sun et al., 2015). Therefore, to make a successful graph matching method, we further consider the learning of node embeddings and derive the proposed framework.

2.2. Proposed model

We propose to not only learn the optimal transport indicating the correspondence between graphs but also simultaneously learn the node embeddings for each graph, which leads to a regularized Gromov-Wasserstein discrepancy. The corresponding optimization problem is

$$\min_{\boldsymbol{X}_s, \boldsymbol{X}_t} \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_s, \boldsymbol{\mu}_t)} \underbrace{\langle \boldsymbol{L}(\boldsymbol{C}_s(\boldsymbol{X}_s), \boldsymbol{C}_t(\boldsymbol{X}_t), \boldsymbol{T}),\ \boldsymbol{T} \rangle}_{\text{Gromov-Wasserstein discrepancy}} + \underbrace{\alpha \langle \boldsymbol{K}(\boldsymbol{X}_s, \boldsymbol{X}_t),\ \boldsymbol{T} \rangle}_{\text{Wasserstein discrepancy}} + \underbrace{\beta R(\boldsymbol{X}_s, \boldsymbol{X}_t)}_{\text{prior information}}. \quad (2)$$

The first term in (2) corresponds to the GW discrepancy defined in (1), which measures the *relational dissimilarity* between the two graphs. The difference here is that the proposed distance matrices consider both the information of observed data and that of embeddings:

$$\boldsymbol{C}_k(\boldsymbol{X}_k) = (1-\alpha)\boldsymbol{C}_k + \alpha \boldsymbol{K}(\boldsymbol{X}_k, \boldsymbol{X}_k), \text{ for } k = s, t. \quad (3)$$

Here $\boldsymbol{K}(\boldsymbol{X}_k, \boldsymbol{X}_k) = [\kappa(\boldsymbol{x}^k_i, \boldsymbol{x}^k_j)] \in \mathbb{R}^{|\mathcal{V}_k| \times |\mathcal{V}_k|}$ is a distance matrix, with element $\kappa(\boldsymbol{x}^k_i, \boldsymbol{x}^k_j)$ that is a function measuring the distance between the node embeddings within the same graph; $\alpha \in [0, 1]$ is a hyperparameter controlling the contribution of embedding-based distance to $\boldsymbol{C}_k(\boldsymbol{X}_k)$.

The second term in (2) represents the Wasserstein discrepancy between the nodes of the two graphs. Similar to the first term, the distance matrix is also derived based on the node embeddings, $i.e.$, $\boldsymbol{K}(\boldsymbol{X}_s, \boldsymbol{X}_t) = [\kappa(\boldsymbol{x}^s_i, \boldsymbol{x}^t_j)] \in \mathbb{R}^{|\mathcal{V}_s| \times |\mathcal{V}_t|}$, and its contribution is controlled by the same hyperparameter α. This term measures the *absolute dissimilarity* between the two graphs, which connects the target optimal transport with node embeddings. By adding this term, the optimal transport minimizes both the Gromov-Wasserstein discrepancy based directly on observed data and the Wasserstein discrepancy based on the embeddings (which are indirectly also a function of the data). Furthermore, the embeddings of different graphs can be learned jointly under the guidance of the optimal transport — the distance between the embeddings of different graphs should be consistent with the relationship indicated by the optimal transport.

Because the target optimal transport is often sparse, purely considering its guidance leads to overfitting or trivial solutions when learning embeddings. To mitigate this problem, the third term in (2) represents a regularization of the embeddings, based on the prior information provided by $\boldsymbol{C}_s$ and $\boldsymbol{C}_t$. We require the embedding-based distance matrices to be close to the observed ones, and $R(\boldsymbol{X}_s, \boldsymbol{X}_t)$ is

$$\sum\nolimits_{k=s,t} L(\boldsymbol{K}(\boldsymbol{X}_k, \boldsymbol{X}_k), \boldsymbol{C}_k) + \underbrace{L(\boldsymbol{K}(\boldsymbol{X}_s, \boldsymbol{X}_t), \boldsymbol{C}_{st})}_{\text{optional}}, \quad (4)$$

where the definition of loss function $L(\cdot, \cdot)$ is the same as that used in (1). Note that if we observe partial correspondences between different graphs, $i.e.$, $\mathcal{E}_{st} = \{(v_i, v_j, w_{ij}) | v_i \in \mathcal{V}_s, v_j \in \mathcal{V}_t\}$, we can calculate a distance matrix for the nodes of different graphs, denoted as $\boldsymbol{C}_{st} \in \mathbb{R}^{|\mathcal{V}_s| \times |\mathcal{V}_t|}$, and require the distance between the embeddings to match with $\boldsymbol{C}_{st}$, as shown in the optional term of (4). This term is available only when $\mathcal{E}_{st}$ is given.

The proposed method unifies (optimal transport-based) graph matching and node embedding in the same framework, and makes them beneficial to each other. For the original GW discrepancy term, introducing the embedding-based distance matrices can suppress the noise in the data-driven distance matrices, improving robustness. Additionally, based on node embeddings, we can calculate the Wasserstein discrepancy between graphs, which further regularizes the target optimal transport directly. When learning node embeddings, the Wasserstein discrepancy term works as the regularizer of node embeddings — the values of the learned optimal transport indicate which pairs of nodes should be close to each other.

3. Learning Algorithm

3.1. Learning optimal transport

Although (2) is a complicated nonconvex optimization problem, we can solve it effectively by alternatively learning the optimal transport and the embeddings. In particular, the proposed method applies nested iterative optimization. In the m-th outer iteration, given current embeddings $\boldsymbol{X}^{(m)}_s$ and $\boldsymbol{X}^{(m)}_t$, we solve the following sub-problem:

$$\min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_s, \boldsymbol{\mu}_t)} \langle \boldsymbol{L}(\boldsymbol{C}_s(\boldsymbol{X}^{(m)}_s), \boldsymbol{C}_t(\boldsymbol{X}^{(m)}_t), \boldsymbol{T}),\ \boldsymbol{T} \rangle + \alpha \langle \boldsymbol{K}(\boldsymbol{X}^{(m)}_s, \boldsymbol{X}^{(m)}_t),\ \boldsymbol{T} \rangle. \quad (5)$$

This sub-problem is still nonconvex because of the quadratic term $\boldsymbol{L}(\boldsymbol{C}_s(\boldsymbol{X}_s^{(m)}), \boldsymbol{C}_t(\boldsymbol{X}_t^{(m)}), \boldsymbol{T}),\ \boldsymbol{T}\rangle$. We solve it iteratively with the help of a proximal point method. Inspired by the method in (Xie et al., 2018), in the n-th inner iteration we update the target optimal transport via

$$\begin{aligned}\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_s,\boldsymbol{\mu}_t)} &\langle \boldsymbol{L}(\boldsymbol{C}_s(\boldsymbol{X}_s^{(m)}), \boldsymbol{C}_t(\boldsymbol{X}_t^{(m)}), \boldsymbol{T}), \boldsymbol{T}\rangle \\ &+ \alpha\langle \boldsymbol{K}(\boldsymbol{X}_s^{(m)}, \boldsymbol{X}_t^{(m)}),\ \boldsymbol{T}\rangle + \gamma \mathrm{KL}(\boldsymbol{T}\|\boldsymbol{T}^{(n)}).\end{aligned} \tag{6}$$

Here, a proximal term based on Kullback-Leibler (KL) divergence, $\mathrm{KL}(\boldsymbol{T}\|\boldsymbol{T}^{(n)}) = \sum_{ij} T_{ij}\log\frac{T_{ij}}{T_{ij}^{(n)}} - T_{ij} + T_{ij}^{(n)}$, is added as a regularizer.

We use projected gradient descent to solve (6), in which both the gradient and the projection are based on the KL metric. When the learning rate is set as $\frac{1}{\gamma}$, the projected gradient descent is equivalent to solving the following optimal transport problem with an entropy regularizer (Benamou et al., 2015; Peyré et al., 2016):

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_s,\boldsymbol{\mu}_t)} \langle \boldsymbol{C}^{(m,n)} - \gamma\log\boldsymbol{T}^{(n)}, \boldsymbol{T}\rangle + \gamma H(\boldsymbol{T}), \tag{7}$$

where $\boldsymbol{C}^{(m,n)} = \boldsymbol{L}(\boldsymbol{C}_s, \boldsymbol{C}_t, \boldsymbol{T}^{(n)}) + \alpha\boldsymbol{K}(\boldsymbol{X}_s^{(m)}, \boldsymbol{X}_t^{(m)}) + \gamma$, and $H(\boldsymbol{T}) = \sum_{i,j} T_{ij}\log T_{ij}$. This problem can be solved via the Sinkhorn-Knopp algorithm (Sinkhorn & Knopp, 1967; Cuturi, 2013) with linear convergence.

In summary, we decompose (5) into a series of updating steps. Each updating step (6) can be solved via projected gradient descent, which is a solution to a regularized optimal transport problem (7). Essentially, the proposed method can be viewed as a special case of successive upper-bound minimization (SUM) (Razaviyayn et al., 2013), whose global convergence is guaranteed:

Proposition 3.1. *Every limit point generated by our proximal point method, i.e., $\lim_{n\to\infty}\boldsymbol{T}^{(n)}$, is a stationary point of the problem (5).*

Note that besides our proximal point method, another method for solving (5) involves replacing the KL-divergence $\mathrm{KL}(\boldsymbol{T}\|\boldsymbol{T}^{(n)})$ in (6) with an entropy regularizer $H(\boldsymbol{T})$ and minimizing an entropic GW discrepancy via iterative Sinkhorn projection (Peyré et al., 2016). However, its performance (*i.e.*, its convergence and numerical stability) is more sensitive to the choice of the hyperparameter γ. The details of our proximal point method, the proof of Proposition 3.1, and its comparison with the Sinkhorn method (Peyré et al., 2016) are shown in the Supplementary Material.

Parameter α controls the influence of node embeddings on the GW discrepancy and the Wasserstein discrepancy. When training the proposed model from scratch, the embeddings $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$ are initialized randomly and thus are unreliable in the beginning. Therefore, we initialize α with a small value and increase it with respect to the number of outer iterations. We apply a simple linear strategy to adjust α: with the maximum number of outer iterations set as M, in the m-th iteration, we set $\alpha_m = \frac{m}{M}$.

Algorithm 1 Gromov-Wasserstein Learning (GWL)

Input: $\{\boldsymbol{C}_s, \boldsymbol{C}_t\}$, $\{\boldsymbol{\mu}_s, \boldsymbol{\mu}_t\}$, β, γ, the dimension D, the number of outer/inner iterations $\{M, N\}$.
Output: $\boldsymbol{X}_s$, $\boldsymbol{X}_t$ and $\widehat{\boldsymbol{T}}$.
Initialize $\boldsymbol{X}_s^{(0)}$, $\boldsymbol{X}_t^{(0)}$ randomly, $\widehat{\boldsymbol{T}}^{(0)} = \boldsymbol{\mu}_s\boldsymbol{\mu}_t^\top$.
For $m = 0 : M-1$
  Set $\alpha_m = \frac{m}{M}$.
  For $n = 0 : N-1$
    Update optimal transport $\widehat{T}^{(m+1)}$ via solving (6).
  Obtain $\boldsymbol{X}_s^{(m+1)}$, $\boldsymbol{X}_t^{(m+1)}$ via solving (8).
$\boldsymbol{X}_s = \boldsymbol{X}_s^{(M)}$, $\boldsymbol{X}_t = \boldsymbol{X}_t^{(M)}$ and $\widehat{\boldsymbol{T}} = \widehat{\boldsymbol{T}}^{(M)}$.
\\ `Graph matching:`
Initialize correspondence set $\mathcal{P} = \emptyset$
For $v_i \in \mathcal{V}_s$
  $j = \arg\max_j \widehat{T}_{ij}$. $\mathcal{P} = \mathcal{P} \cup \{(v_i \in \mathcal{V}_s, v_j \in \mathcal{V}_t)\}$.

3.2. Updating embeddings

Given the optimal transport, $\widehat{\boldsymbol{T}}^{(m)}$, we update the embeddings by solving the following optimization problem:

$$\min_{\boldsymbol{X}_s,\boldsymbol{X}_t} \alpha_m\langle \boldsymbol{K}(\boldsymbol{X}_s, \boldsymbol{X}_t),\ \widehat{\boldsymbol{T}}^{(m)}\rangle + \beta R(\boldsymbol{X}_s, \boldsymbol{X}_t). \tag{8}$$

This problem can be solved effectively by (stochastic) gradient descent. In summary, the proposed learning algorithm is shown in Algorithm 1.

3.3. Implementation details and analysis

Distance matrix The distance matrix plays an important role in our Gromov-Wasserstein learning framework. For a graph, the data-driven distance matrix should reflect its structure. Based on the fact that the counts of interactions in many real-world graphs is characterized by Zipf's law (Powers, 1998), we treat the counts as the weights of edges and define the element of the data-driven distance matrix as

$$c_{ij}^k = \begin{cases}\frac{1}{w_{ij}+1}, & (v_i, v_j) \in \mathcal{E}_k, \\ 1, & (v_i, v_j) \notin \mathcal{E}_k,\end{cases} \quad \text{for } k = s, t. \tag{9}$$

This definition assigns a short distance to pairs of nodes with many interactions. Additionally, we hope that the embedding-based distance matrix can fit the data-driven distance matrix easily. In the following experiments, we test two kinds of embedding-based distance: 1) Cosine-based distance: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \exp(-\sigma(1 - \frac{\boldsymbol{x}_i^\top\boldsymbol{x}_j}{\|\boldsymbol{x}_i\|_2\|\boldsymbol{x}_j\|_2}))$. 2) Radial basis function (RBF)-based distance: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{\sigma^2})$. When applying the cosine-based distance, we choose $\sigma = 10$ such that the maximum $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$

approaches to 1. When applying the RBF-based distance, we choose $\sigma = D$. The following experiments show that these two distances work well in various matching tasks.

Complexity and Scalability When learning optimal transport, one of the most time-consuming steps is computing the loss matrix $\boldsymbol{L}(\boldsymbol{C}_s, \boldsymbol{C}_t, \boldsymbol{T})$, which involves a tensor-matrix multiplication. Fortunately, as shown in (Peyré et al., 2016) when the loss function $L(a, b)$ can be written as $L(a, b) = f_1(a) + f_2(b) - h_1(a)h_2(b)$ for functions (f_1, f_2, h_1, h_2), which is satisfied by our MSE/KL loss, the loss matrix can be calculated as $\boldsymbol{L}(\boldsymbol{C}_s, \boldsymbol{C}_t, \boldsymbol{T}) = f_1(\boldsymbol{C}_s)\boldsymbol{\mu}_s \mathbf{1}_{|\mathcal{V}_t|}^\top + \mathbf{1}_{|\mathcal{V}_s|}\boldsymbol{\mu}_t^\top f_2(\boldsymbol{C}_t)^\top - h_1(\boldsymbol{C}_s)\boldsymbol{T}h_2(\boldsymbol{C}_t)^\top$. Because $\boldsymbol{T}$ tends to be sparse quickly during the learning process, the computational complexity of $L(a, b) = f_1(a)+f_2(b)-h_1(a)h_2(b)$ is $\mathcal{O}(V^3)$, where $V = \max\{|\mathcal{V}_s|, |\mathcal{V}_t|\}$. For D-dimensional node embeddings, the complexity of the embedding-based distance matrix $\boldsymbol{K}(\boldsymbol{X}_s, \boldsymbol{X}_t)$ is $\mathcal{O}(V^2D)$. Additionally, we can apply the inexact proximal point method (Xie et al., 2018; Chen et al., 2018a), running one-step Sinkhorn-Knopp projection in each inner iteration. Therefore, the complexity of learning optimal transport is $\mathcal{O}(V^2D + NV^3)$. When learning node embeddings, we can apply stochastic gradient descent to solve (8). In our experiments, we select the size of the node batch as $B \ll V$ and the objective function of (8) converges quickly after a few epochs. Therefore, the computational complexity of the embedding-based distance sub-matrix is just $\mathcal{O}(B^2D)$, which may be ignored compared to that of learning optimal transport. In summary, the overall complexity of our method is $\mathcal{O}(M(V^2D + NV^3))$, and both the learning of optimal transport and that of node embeddings can be done in parallel on GPUs.

Note that the proposed method has lower complexity than many existing graph matching methods. For example, the GRAAL and its variants (Malod-Dognin & Pržulj, 2015) have $\mathcal{O}(V^5)$ complexity, which is much slower than the proposed method. Additionally, the complexity of our method is independent of the number of edges (denoted as $E = \max\{|\mathcal{E}_s|, |\mathcal{E}_t|\}$). Compared to other well-known alternatives, $e.g.$, NETAL (Neyshabur et al., 2013) with $\mathcal{O}(V \log V + E^2 + EV \log V)$, our method has at least comparable complexity for dense graphs ($E \gg V$).

4. Related Work

Gromov-Wasserstein learning Gromov-Wasserstein discrepancy extends optimal transport (Villani, 2008) to the case when the target domains are not registered well. It can also be viewed as a relaxation of Gromov-Hausdorff distance (Mémoli, 2008; Bronstein et al., 2010) when pairwise distance between entities is defined. The GW discrepancy is suitable for solving matching problems like shape and object matching (Mémoli, 2009; 2011). Besides graphics and computer vision, recently its potential for other applications has been investigated, $e.g.$, matching vocabulary sets between different languages (Alvarez-Melis & Jaakkola, 2018) and matching weighted directed networks (Chowdhury & Mémoli, 2018). The work in (Peyré et al., 2016) considers the Gromov-Wasserstein barycenter and proposes a fast Sinkhorn projection-based algorithm to compute GW discrepancy (Cuturi, 2013). Similar to our method, the work in (Vayer et al., 2018) proposes a fused Gromov-Wasserstein distance, combining GW discrepancy with Wasserstein discrepancy. However, it does not consider the learning of embeddings and requires the distance between the entities in different domains to be known, which is inapplicable to matching problems. In (Bunne et al., 2018), an adversarial learning method is proposed to learn a pair of generative models for incomparable spaces, which uses GW discrepancy as the objective function. This method imposes an orthogonal assumption on the transformation between the sample and its embedding; it is designed for fuzzy matching between distributions, rather than the graph matching task that requires point-to-point correspondence.

Graph matching Graph matching has been studied extensively, with a wide range of applications. Focusing on protein-protein interaction (PPI) networks, many methods have been proposed, including methods based on local neighborhood information like GRAAL (Kuchaiev et al., 2010), and its variants MI-GRAAL (Kuchaiev & Pržulj, 2011) and L-GRAAL (Malod-Dognin & Pržulj, 2015); as well as methods based on global structural information, like IsoRank (Singh et al., 2008), MAGNA++ (Vijayan et al., 2015), NETAL (Neyshabur et al., 2013), HubAlign (Hashemifar & Xu, 2014) and WAVE (Sun et al., 2015). Among these methods, MAGNA++ and WAVE consider both edge and node information. Besides bioinformatics, network alignment techniques are also applied to computer vision (Jun et al., 2017; Yu et al., 2018), document analysis (Bayati et al., 2009) and social network analysis (Zhang & Philip, 2015). For small graphs, $e.g.$, the graph of feature points in computer vision, graph matching is often solved as a quadratic assignment problem (Yan et al., 2015). For large graphs, $e.g.$, social networks and PPI networks, existing methods either depend on a heuristic searching strategy or leverage domain knowledge for specific cases. None of these methods consider graph matching and node embedding jointly from the viewpoint of Gromov-Wasserstein discrepancy.

Node embedding Node embedding techniques have been widely used to represent and analyze graph/network structures. The representative methods include LINE (Tang et al., 2015), Deepwalk (Perozzi et al., 2014), and node2vec (Grover & Leskovec, 2016). Most of these embedding methods first generate sequential observations of nodes through a random-walk procedure, and then learn the embeddings by maximizing the coherency between each

observation and its context (Mikolov et al., 2013). The distance between the learned embeddings can reflect the topological structure of the graph. More recently, many new embedding methods have been proposed, *e.g.*, the anonymous walk embedding in (Ivanov & Burnaev, 2018) and the mixed membership word embedding (Foulds, 2018), which help to improve the representations of complicated graphs and their nodes. However, none of these methods consider jointly learning embeddings for multiple graphs.

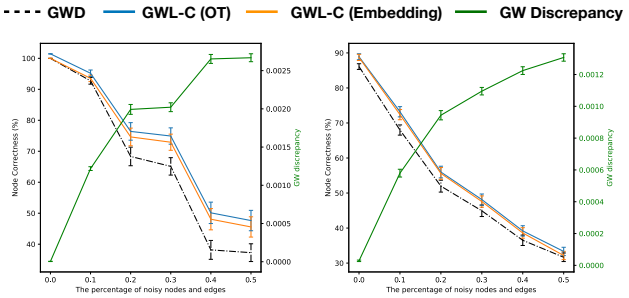
5. Experiments

We apply the Gromov-Wasserstein learning (GWL) method to both synthetic and real-world matching tasks, and compare it with state-of-the-art methods. In our experiments, we set hyperparameters as follows: the number of outer iterations is $M = 30$, the number of inner iteration is $N = 200$, $\gamma = 0.01$ and $L(\cdot, \cdot)$ is the MSE loss. We tried βs in $\{0, 1, 10, 100, 1000\}$ and the β in $[1, 100]$ achieves stable performance. Therefore, we empirically set $\beta = 10$. When solving (8), we use Adam (Kingma & Ba, 2014) with learning rate 0.001 and set the number of epochs to 5, and the size of batches as 100. The proposed method based on cosine and RBF distances are denoted **GWL-C** and **GWL-R**, respectively. Additionally, to highlight the benefit from joint graph matching and node-embedding learning, we consider a baseline that purely minimizes GW discrepancy based on data-driven distance matrices (denoted as **GWD**). The code is available on https://github.com/HongtengXu/gwl.

5.1. Synthetic data

We verify the feasibility of our GWL method by first considering two kinds of synthetic datasets. The graphs in the first dataset imitate K-NN graphs with certain randomness, which is common in practical data science. The graphs in the second dataset yield to the Barabási-Albert (BA) model (Barabási et al., 2016), which matches the statistics of real-world networks well. For the K-NN graph dataset, we simulate the source graph $G(\mathcal{V}_s, \mathcal{E}_s)$ as follows: for each $v_i \in \mathcal{V}_s$, we select $K \sim \text{Poisson}(0.1 \times |\mathcal{V}_s|)$ nodes randomly from $\mathcal{V}_s \setminus v_i$, denoted as $\{v_i^k\}_{k=1}^K$. For each selected edge (v_i, v_i^k), there are $w \sim \text{Poisson}(10)$ interactions between these two nodes. Accordingly, $\mathcal{E}_s$ is the union of all simulated $\{(v_i, v_i^k, w)\}_{i,k}$. The target graph $G(\mathcal{V}_t, \mathcal{E}_t)$ is constructed by first adding $q\%$ noisy nodes to the source graph, *i.e.*, $|\mathcal{V}_t| = (1 + q\%)|\mathcal{V}_s|$, and then generating $q\%$ noisy edges between the nodes in $\mathcal{V}_t$ via the simulation method mentioned above, *i.e.*, $|\mathcal{E}_t| = (1 + q\%)|\mathcal{E}_s|$. Similarly, for the BA graph dataset, we first simulate the source graph with $|\mathcal{V}_s|$ nodes, and then simulate the target graph via adding $q\%|\mathcal{V}_s|$ more nodes and corresponding edges.

We set $|\mathcal{V}_s| \in \{50, 100\}$ and $q \in \{0, 10, 20, 30, 40, 50\}$. For each configuration, we simulate the source graph and the target one in 100 trials. For each trial, we apply our method (and its baseline GWD) to match the graphs and calculate *node correctness* as our measurement: Given the learned correspondence set $\mathcal{P}$ and the ground truth set of correspondences $\mathcal{P}_{real}$, we calculate percent node correctness as $NC = \frac{|\mathcal{P} \cap \mathcal{P}_{real}|}{|\mathcal{P}|} \times 100\%$. To analyze the rationality of the learned node embeddings, we construct $\mathcal{P}$ in two ways: for each $v_i^s \in \mathcal{V}_s$, we find its matched node $v_j^t \in \mathcal{V}_t$ via (i) $\arg\max_j \widehat{T}_{ij}$ (as shown in line 13 of Algorithm 1) or (ii) $\arg\min_j \kappa(\boldsymbol{x}_i^s, \boldsymbol{x}_j^t)$. Additionally, the corresponding GW discrepancy is calculated as well. Assuming that the results in different trials are Gaussian distributed, we calculate the 95% confidence interval for each measurement.



(a) K-NN: $|\mathcal{V}_s| = 50$ (b) BA: $|\mathcal{V}_s| = 50$

Figure 2. The performance of our method on synthetic data.

Figure 2 visualizes the performance of our GWL-C method and its baseline GWD when $|\mathcal{V}_s| = 50$. More results are in the Supplementary Material. When the target graph is identical to the source one (*i.e.*, $q = 0$), the proposed Gromov-Wasserstein learning framework can achieve almost 100% node correctness, and the GW discrepancy approaches zero. With the increase of q, the noise in the target graph becomes serious, and the GW discrepancy increases accordingly. It means that the GW discrepancy reflects the dissimilarity between the graphs indeed. Although the GWD is comparable to our GWL-C in the case with low noise level, it becomes much worse when $q > 20$. This phenomenon supports our claim that learning node embeddings can improve the robustness of graph matching. Moreover, we find that the node correctness based on the optimal transport (blue curves) and that based on the embeddings (orange curves) are almost the same. This demonstrates that the embeddings of different graphs are on the same manifold, and their distances indicate the correspondences between graphs.

5.2. MC3: Matching communication networks

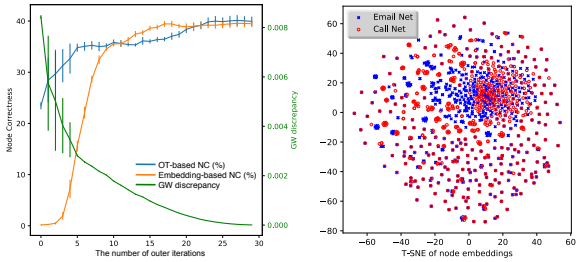
MC3 is a dataset used in the Mini-Challenge 3 of VAST Challenge 2018, which records the communication behavior among a company's employees on different networks.[1] The communications are categorized into two types: phone calls

[1] http://vacommunity.org/VAST+Challenge+2018+MC3

and emails between employees. According to the types of the communications, we obtain two networks, denoted as *CallNet* and *EmailNet*. Because an employee has two independent accounts in these two networks, we aim to link the accounts belonging to the same employee. We test our method on a subset of the MC3 dataset, which contains 622 employees and their communications through phone calls and emails. In this subset, for each selected employee there is at least one employee in a network (either CallNet or EmailNet) having over 10 times communications with him/her, which ensures that each node has at least one reliable edge. Additionally, for each network, we can control the density of its edge by thresholding the count of interactions. When we only keep the edges corresponding to the communications happening more than 8 times, we obtain two sparse graphs: the CallNet contains $1,228$ edges and the EmailNet contains $1,235$ edges. When we keep all the communications and the corresponding edges, we obtain two dense graphs, the CallNet contains $141,846$ edges and the EmailNet contains $115,782$ edges. Generally, experience indicates that matching dense graphs is much more difficult than matching sparse ones.

We compare our methods (GWL-R and GWL-C) with well-known graph matching methods: the graduated assignment algorithm (GAA) (Gold & Rangarajan, 1996), the low-rank spectral alignment (LRSA) (Nassar et al., 2018), TAME (Mohammadi et al., 2017), GRAAL[2], MI-GRAAL[3], MAGNA++[4], HugAlign and NETAL.[5] These alternatives achieve the state-of-the-art performance on matching large-scale graphs, $e.g.$, protein networks. Table 1 lists the matching results obtained by the different methods.[6] For the alternative methods, their best results in 10 trials are listed. We can find that their performance on sparse and dense graphs is inconsistent. For example, GRAAL works almost as well as our GWL-R and GWL-C for sparse graphs, but its matching result becomes much worse for dense graphs. For the baseline GWD, it is inferior to most graph-matching methods on node correctness, because it purely minimizes the GW discrepancy based on the information of pairwise interactions ($i.e.$, edges). Additionally, GWD merely relies on data-driven distance matrices, which is sensitive to the noise in the graphs. However, when we take node embeddings (with dimension $D = 100$) into account, the proposed GWL-R and GWL-C outperform GWD and other considered approaches consistently, on both sparse and dense graphs.

To demonstrate the convergence and the stability of our method, we run GWD, GWL-R and GWL-C in 10 trials with different initialization. For each method, its node correctness is calculated based on optimal transport and the embedding-based distance matrix. The 95%-confidence interval of the node correctness is estimated as well, as shown in Table 1. We find that the proposed method has good stability and outperforms other methods with high confidence. Figure 3(a) visualizes the GW discrepancy and the node correctness with respect to the number of outer iterations; the 95%-confidence intervals are shown as well. In Figure 3(a), we find that the GW discrepancy decreases and the two kinds of node correctness increase accordingly and become consistent with the increase of iterations, which means that the embeddings we learn and their distances indeed reflect the correspondence between the two graphs. Figure 3(b) visualizes the learned embeddings with the help of t-SNE (Maaten & Hinton, 2008). We find that the learned node embeddings of different graphs are on the same manifold and the overlapped embeddings indicate matched pairs.



(a) Stability and convergence (b) Learned embeddings

Figure 3. Visualization of typical experimental results.

Table 1. Communication network matching results.

Method	Call→Email (Sparse)	Call→Email (Dense)
	Node Correctness (%)	Node Correctness (%)
GAA	34.22	0.53
LRSA	38.20	2.93
TAME	37.39	2.67
GRAAL	39.67	0.48
MI-GRAAL	35.53	0.64
MAGNA++	7.88	0.09
HugAlign	36.21	3.86
NETAL	36.87	1.77
GWD	23.16±0.46	1.77±0.22
GWL-R	39.64±0.57	3.80±0.23
GWL-C	**40.45**±0.53	**4.23**±0.27

5.3. MIMIC-III: Procedure recommendation

Besides typical graph matching, our method has potential for other applications, like recommendation systems. Such systems recommend items to users according to the distance/similarity between their embeddings. Traditional methods (Rendle et al., 2009; Chen et al., 2018b) learn the embeddings of users and items purely based on their interactions. Recent work (Monti et al., 2017; Ying et al., 2018) shows that considering the user network and/or item net-

[2]http://www0.cs.ucl.ac.uk/staff/natasa/GRAAL.

[3]http://www0.cs.ucl.ac.uk/staff/natasa/MI-GRAAL.

[4]https://www3.nd.edu/~cone/MAGNA++.

[5]http://ttic.uchicago.edu/~hashemifar.

[6]For GWD, GWL-R and GWL-C, here we show the node correctness calculated based on the learned optimal transport.

work is beneficial to improve recommendation results. Such a strategy is also applicable to our Gromov-Wasserstein learning framework: given the network of users, the network of items, and the observed interactions between them ($i.e.$, partial correspondences between the graphs), we learn the embeddings of users and items and the optimal transport between them via minimizing the GW discrepancy between the networks. Because the learned embeddings are on the same manifold, we can calculate the distance between a user and an item directly via the cosine-based distance or the RBF-based distance. Accordingly, we recommend each user with the items with shortest distances. For our method, the only difference between the recommendation task and previous graph matching task is that we observed some interactions, $i.e.$, the w_{ij} between source node $v_i \in \mathcal{V}_s$ and target node $v_j \in \mathcal{V}_t$. In such a situation, we take the optional regularizer in (4) into account. Based on observed w_{ij}'s, the elements of the $\boldsymbol{C}_{st}$ in (4) are calculated via (9).

We test the feasibility of our method on the MIMIC-III dataset (Johnson et al., 2016), which contains patient admissions in a hospital. Each admission is represented as a sequence of ICD (International Classification of Diseases) codes of the diseases and the procedures. The diseases (procedures) appearing in the same admission construct the interactions of the disease (procedure) graph. We aim to recommend suitable procedures for patients, according to their disease characteristics. To achieve this, we learn the embeddings of the ICD codes for the diseases and the procedures with the help of various methods, and measure the distance between the embeddings. We compare the proposed GWL method with the following baselines: i) treating the admission sequences as sentences and learning the embeddings of ICD codes via traditional word embedding methods like **Word2Vec** (Mikolov et al., 2013) and **GloVe** (Pennington et al., 2014); ii) the distilled Wasserstein learning (**DWL**) method in (Xu et al., 2018), which trains the embeddings from scratch or fine-tunes Word2Vec's embeddings based on a Wasserstein topic model; and iii) the GWD method that minimizes the GW discrepancy purely based on the data-driven distance matrices, and then learns the embeddings regularized by the learned optimal transport. The GWD method is equivalent to applying our GWL method and setting the number of outer iterations $M = 1$. For the GWD method, we also consider the cosine- and RBF-based distances when learning embeddings, denoted as **GWD-C** and **GWD-R**, respectively.

For fairness of comparison, we use a subset of the MIMIC-III dataset provided by (Xu et al., 2018), which contains $11,086$ patient admissions, corresponding to 56 diseases and 25 procedures. For all the methods, we use 50% of the admissions for training, 25% for validation, and the remaining 25% for testing. In the testing phase, for the i-th admission, $i = 1, ..., I$, we may recommend a list of procedures with length L, denoted as E_i, based on its diseases and evaluate recommendation results based on the ground truth list of procedures, denoted as T_i. Given $\{E_i, T_i\}$, we calculate the top-L precision, recall and F1-score as follows: $P = \sum_{i=1}^{I} P_i = \sum_{i=1}^{I} \frac{|E_i \cap T_i|}{|E_i|}$, $R = \sum_{i=1}^{I} R_i = \sum_{i=1}^{I} \frac{|E_i \cap T_i|}{|T_i|}$, $F1 = \sum_{i=1}^{I} \frac{2P_iR_i}{P_i+R_i}$. Table 2 shows the results of various methods with $L = 1$ and 5. We find that our GWL method outperforms the alternatives, especially on the top-1 measurements.

Table 2. Top-N procedure recommendation results.

Method	Top-1 (%)			Top-5 (%)		
	P	R	F1	P	R	F1
Word2Vec	39.95	13.27	18.25	28.89	46.98	32.59
GloVe	32.66	13.01	17.22	27.93	44.79	31.47
DWL (Scratch)	37.89	12.42	17.16	27.39	43.81	30.81
DWL (Finetune)	40.00	13.76	18.71	30.59	**48.56**	34.28
GWD-R	46.29	17.01	22.32	31.82	43.81	33.77
GWD-C	43.16	15.79	20.77	31.42	42.99	33.25
GWL-R	46.20	16.93	22.22	32.03	44.75	34.18
GWL-C	**47.46**	**17.25**	**22.71**	**32.09**	45.64	**34.31**

We analyze the learned optimal transport between diseases and procedures from a clinical viewpoint. In particular, we normalize the transport matrix, ensuring its maximum value is 1. For each disease, we find the corresponding procedures i) with the maximum optimal transports and ii) with $\widehat{T}_{ij} > 0.15$. We asked two clinical researchers to check the pairs we find; they confirmed that for over 77.42% of the pairs, either the procedures are clearly related to the treatments of the diseases, or the procedures clearly lead to the diseases as side effects or complications (other relationships may be less clear, but are implied by the data), $e.g.$, "(dV3001) Single liveborn, born in hospital, delivered by cesarean section $\leftrightarrow$ (p640) Circumcision". The learned optimal transport, and all pairs of ICD codes and their evaluation results are shown in the Supplementary Material.

6. Conclusions and Future Work

We have proposed a Gromov-Wasserstein learning method to unify graph matching and the learning of node embeddings into a single framework. We show that such joint learning is beneficial to each of the objectives, obtaining superior performance in various matching tasks. In the future, we plan to extend our method to multi-graph matching (Yan et al., 2016), which may be related to Gromov-Wasserstein barycenter (Peyré et al., 2016) and its learning method. Additionally, to improve the scalability of our method, we will explore new Gromov-Wasserstein learning algorithms.

Acknowledgments This research was supported in part by DARPA, DOE, NIH, ONR and NSF. We thank Dr. Matthew Engelhard and Rachel Draelos for evaluating our results. We also thank Wenlin Wang for helpful discussions.

References

Altschuler, J., Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. *arXiv preprint arXiv:1705.09634*, 2017.

Alvarez-Melis, D. and Jaakkola, T. Gromov-Wasserstein alignment of word embedding spaces. In *EMNLP*, pp. 1881–1890, 2018.

Barabási, A.-L. et al. *Network science*. Cambridge university press, 2016.

Bayati, M., Gerritsen, M., Gleich, D. F., Saberi, A., and Wang, Y. Algorithms for large, sparse network alignment problems. In *ICDM*, pp. 705–710, 2009.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Bronstein, A. M., Bronstein, M. M., Kimmel, R., Mahmoudi, M., and Sapiro, G. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2-3):266–286, 2010.

Bunne, C., Alvarez-Melis, D., Krause, A., and Jegelka, S. Learning generative models across incomparable spaces. *NeurIPS Workshop on Relational Representation Learning*, 2018.

Chen, L., Dai, S., Tao, C., Zhang, H., Gan, Z., Shen, D., Zhang, Y., Wang, G., Zhang, R., and Carin, L. Adversarial text generation via feature-mover's distance. In *NIPS*, pp. 4671–4682, 2018a.

Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., and Zha, H. Sequential recommendation with user memory networks. In *WSDM*, pp. 108–116, 2018b.

Chowdhury, S. and Mémoli, F. The Gromov-Wasserstein distance between networks and stable network invariants. *arXiv preprint arXiv:1808.04337*, 2018.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, 2004.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, pp. 2292–2300, 2013.

Foulds, J. Mixed membership word embeddings for computational social science. In *AISTATS*, pp. 86–95, 2018.

Gold, S. and Rangarajan, A. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.

Grave, E., Joulin, A., and Berthet, Q. Unsupervised alignment of embeddings with Wasserstein Procrustes. *arXiv preprint arXiv:1805.11222*, 2018.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *KDD*, pp. 855–864, 2016.

Hashemifar, S. and Xu, J. Hubalign: An accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, 30(17):i438–i444, 2014.

Ivanov, S. and Burnaev, E. Anonymous walk embeddings. In *ICML*, 2018.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Jun, S.-H., Wong, S. W., Zidek, J., and Bouchard-Côté, A. Sequential graph matching with sequential monte carlo. In *AISTATS*, pp. 1075–1084, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kuchaiev, O. and Pržulj, N. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011.

Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., and Pržulj, N. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, pp. rsif20100063, 2010.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov): 2579–2605, 2008.

Malod-Dognin, N. and Pržulj, N. L-GRAAL: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31(13): 2182–2189, 2015.

Mémoli, F. Gromov-Hausdorff distances in Euclidean spaces. In *CVPR Workshops*, pp. 1–8, 2008.

Mémoli, F. Spectral Gromov-Wasserstein distances for shape matching. In *ICCV Workshops*, pp. 256–263, 2009.

Mémoli, F. Gromov-Wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Mohammadi, S., Gleich, D. F., Kolda, T. G., and Grama, A. Triangular alignment TAME: A tensor-based approach for higher-order network alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 14(6):1446–1458, 2017.

Monti, F., Bronstein, M., and Bresson, X. Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS*, pp. 3697–3707, 2017.

Myronenko, A. and Song, X. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

Nassar, H., Veldt, N., Mohammadi, S., Grama, A., and Gleich, D. F. Low rank spectral network alignment. In *WWW*, pp. 619–628, 2018.

Neyshabur, B., Khadem, A., Hashemifar, S., and Arab, S. S. NETAL: A new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29(13):1654–1662, 2013.

Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *KDD*, pp. 701–710, 2014.

Peyré, G., Cuturi, M., and Solomon, J. Gromov-Wasserstein averaging of kernel and distance matrices. In *ICML*, pp. 2664–2672, 2016.

Powers, D. M. Applications and explanations of zipf's law. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pp. 151–160, 1998.

Razaviyayn, M., Hong, M., and Luo, Z.-Q. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pp. 452–461, 2009.

Sharan, R. and Ideker, T. Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4):427, 2006.

Singh, R., Xu, J., and Berger, B. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 2008.

Sinkhorn, R. and Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Sun, Y., Crawford, J., Tang, J., and Milenković, T. Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *International Workshop on Algorithms in Bioinformatics*, pp. 16–39, 2015.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. Line: Large-scale information network embedding. In *WWW*, pp. 1067–1077, 2015.

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.

Vijayan, V., Saraph, V., and Milenković, T. MAGNA++: Maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, 31(14): 2409–2411, 2015.

Villani, C. *Optimal transport: Old and new*, volume 338. Springer Science & Business Media, 2008.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for Wasserstein distance. *arXiv preprint arXiv:1802.04307*, 2018.

Xu, H., Wang, W., Liu, W., and Carin, L. Distilled Wasserstein learning for word embedding and topic modeling. In *NIPS*, pp. 1723–1732, 2018.

Yan, J., Xu, H., Zha, H., Yang, X., Liu, H., and Chu, S. A matrix decomposition perspective to multiple graph matching. In *ICCV*, pp. 199–207, 2015.

Yan, J., Cho, M., Zha, H., Yang, X., and Chu, S. M. Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1228–1242, 2016.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. *arXiv preprint arXiv:1806.01973*, 2018.

Yu, T., Yan, J., Wang, Y., Liu, W., et al. Generalizing graph matching beyond quadratic assignment model. In *NIPS*, pp. 861–871, 2018.

Zhang, J. and Philip, S. Y. Multiple anonymized social networks alignment. In *ICDM*, pp. 599–608, 2015.