A Regularized Wasserstein Framework for Graph Kernels

Asiri Wijesinghe, Qing Wang, and Stephen Gould School of Computing, Australian National University, Canberra, Australia

Grégoire Béchade

January 16 2025

Introduction

1. A measure to compare graph similarity
2. Numerical experiments

From graphs to distributions

In order to compare graphs with Wasserstein distance, we need to associate each graph with a distribution. Let :

- $\xi_f : V \to \mathbb{R}^m$ a feature embedding of the nodes
- $\xi_s : V \to \mathbb{R}^k$ a structure embeding of the nodes

$p = \sum_{i=0}^{n} \mu_i \delta(\xi_f(v_i), \xi_s(v_i))$ can be seen as a distribution on $\mathbb{R}^{m+k}$.

Defining the Wasserstein problem

To solve a Wasserstein problem, one need to define a cost function :

$C^V(i,j) = \|(\xi_f(v_i), \xi_f(v_j))\|^2$

With $\xi_f(v_i)$ the feature embedding of the node v_i :

$\xi_f(v_i) = [x_i, \Delta(x_i)] \in \mathbb{R}^{2m}$ and $\Delta(x_i)$ the local variation of the node x_i.

LW : A Regularization to Preserve Neighborhood Similarity

LW distance :

$$LW(\mu, \nu) = \min_{\gamma} \left\langle \gamma, C^N \right\rangle_F + \Theta_\omega(\gamma)$$

$C^N(i, j) = d_s(e_i, e_j)$, with e_i and e_j the embeddings of the nodes v_i and v_j

$\Theta_w(\gamma) = \lambda_\mu \Gamma_\mu(\gamma) + \lambda_\nu \Omega_\nu(\gamma) = \frac{\rho}{2} \|\gamma\|_F{}^2$

LW : A Regularization to Preserve Neighborhood Similarity

$\hat{e}_i$: mean of the embeddings of the nodes connected to v_i in γ :

$$\hat{e}_i^{\mu} = \frac{\sum_{j=1}^{n_2} \gamma(i,j) e_j^{\nu}}{\sum_{j=1}^{n_2} \gamma(i,j)}.$$

Source regularization : $\Omega_{\mu}(\gamma) = \frac{1}{n_1^2} \sum_{i,j} a_{i,j} \left\| \hat{e}_i^{\mu} - \hat{e}_j^{\mu} \right\|^2$, with $a_{i,j}$ the adjacency matrix of G_1.

Target regularization : Defined the same way.

GW : A Regularization to Preserve Pairwise Similarity

- **Pairwise similarity :**
  - Cost matrix for graph G : $C^P(i,j) = d_s(e_i, e_j)$.
  - Pairwise similarity between G_1 and G_2 :

$$L_2(C_1^P(i,j), C_2^P(k,l)) = \left|C_1^P(i,j) - C_2^P(k,l)\right|^2$$

- **Gromov-Wasserstein (GW) distance :**

$$GW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} \left\langle \gamma, L_2(C_1^P, C_2^P) \otimes \gamma \right\rangle_F - \lambda_g \Theta_g(\gamma)$$

  - $\Theta_g(\gamma)$: Kullback-Leibler divergence with prior γ'.
  - Regularization improves convergence despite GW not being convex.

solving this optimization problem with the SCG algorithm

- NP-hard problem.
- SCG algorithm that iterates by using the gradient of the objective function as the cost matrix in a Sinkhorn-Kopp algorithm.
- Theoretical guarantees of the convergence.

Using This distance to perform graph classification

The authors use kernel methods to classify graphs :

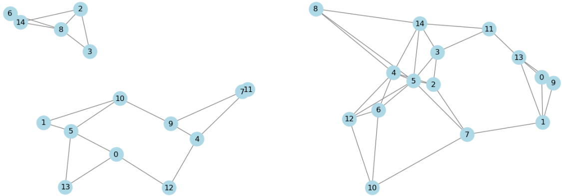
$$K(G_1, G_2) = e^{(-\eta RW(G_1,G_2))}$$

Which is not positive definite but is seen as the noisy observation of a positive definite kernel.

Data used for numerical experiments

- Task : classification of 2-nn and 3-nn graphs.

- Train set : 400 graphs
- Test set : 100 graphs
- Balanced classes



(a) Example of a graph with k = 2

(b) Example of a graph with k = 3

Figure – Example of two graphs of each class

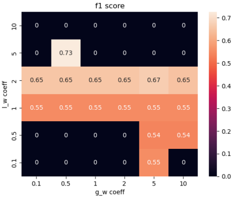
Impact of the different parameters

I have studied the impact of the different parameters on this simple task.

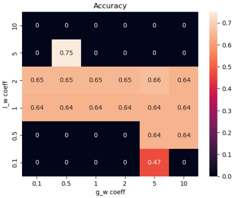
	Time (min)	F1-score	Accuracy
Default	47	0.65	0.73
$\beta_2/\lambda_g = 0$	25	0.78	0.76
$\rho = 0$	62	0.63	0.71
$\lambda_\mu = 0$	58	0.65	0.71
$\lambda_\nu = 0$	32	0.65	0.72
$\lambda_g = 0$	38	0.61	0.69
Sinkhorn algorithm	11	0	0.5
Random Forest	2.10^{-2}	0.63	0.67
SVC Gaussian kernel	0	0.60	0.66
KNN	1	0.7	0.68

Table – Performances of the classifier with different variations

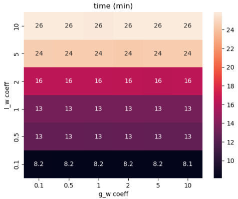
Impact of the two regularization terms



(a) F1-score



(b) Accuracy



(c) Computational time

Figure – Results of the classifier with variations of the two regularization terms

Conclusion

- An complete way to determine the similarity between graphs.
- Results of the paper not reproduced on a single classification task.