Projet *graph neural net*

Présentation - MAP583

Nathanaël Cuvelle-Magar, Mathieu Guesmi, Yongyi Hu

8 mars 2022

Plan

Introduction - Quadratic Assignment Problem

Notations

- Given two finite sets P and L
- Weight function $w : P \times P \longrightarrow \mathbb{R}$
- Distance function $d : L \times L \longrightarrow \mathbb{R}$

Objective

$$\min_{\substack{f:P \longrightarrow L \\ bijection}} \sum_{a,b \in P} w(a,b) \cdot d(f(a), f(b)) \tag{1}$$

Original Situation

- P - Facilities ; L - Locations
- $w(a,b)$ - Amount of goods to transfer
- $d(f(a), f(b))$ - Distance between the factories

Introduction - Network Alignment

Problem

Find the "best" mapping between the vertices of two graphs.

Objective

$$\max_{\sigma \in S_n} \sum_{i,j \in [n]} A_{ij} \cdot B_{\sigma(i),\sigma(j)} \tag{2}$$

where A, B are seperately the adjacency matrix for two graphs of dimension $n \times n$, S_n n-permutation set.

Relation with QAP

Comparing the Eqn. (1) and (2), the Network Aligment problem is a special case of Quadratic Alignment Problem under this setting.

Plan

Architecture du code pour la résolution du *QAP*

Idée générale

- Utilisation d'un *GNN* siamois pour encoder les noeuds des graphes à aligner.
- En multipliant les plongements obtenus pour les deux graphes, on obtient une matrice de similarité sur les noeuds.
- On obtient finalement une permutation sur les indexes en résolvant un problème d'affectation.

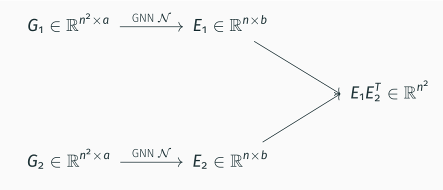
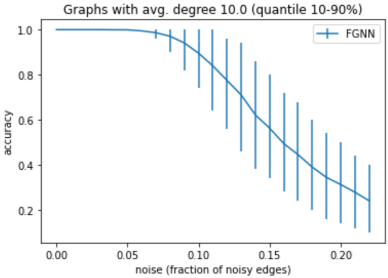


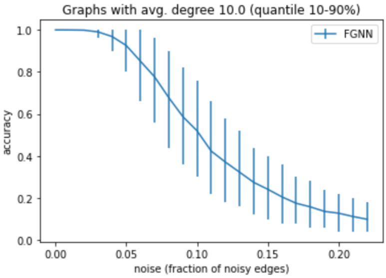
Figure – [AL21]

Exécution sur Google Colab

Apports

- Rédaction d'un notebook pour exécution avec les GPUs disponibles sur Google Colab.
- Adaptation du code d'un notebook existant pour l'affichage des résultats.



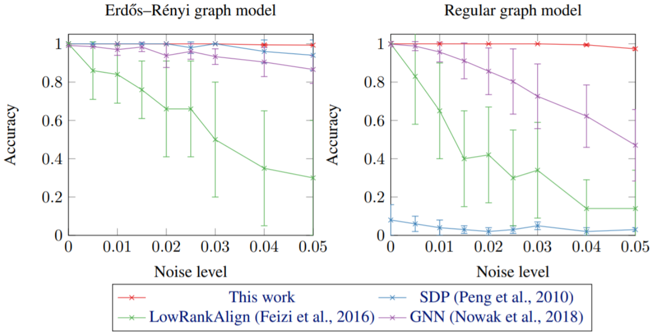


Plan

Comparaisons between Regular and Erdos Reyni

Observations

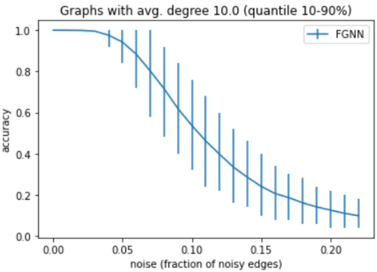
- Regular generated graphs are more complicated to be aligned that Erdos Reyni ones for our problem.
- It does not seem to depend to the algorithm.



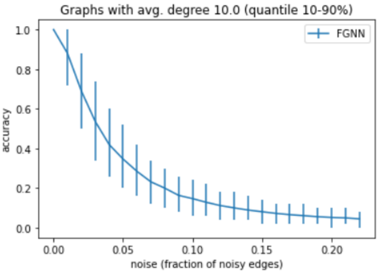
Changing the graph generator

Observations

- We observe the previous phenomenon with our algorithm.
- The training on Regular generated graphs is more generalizable than that on Erdos-Reyni ones.



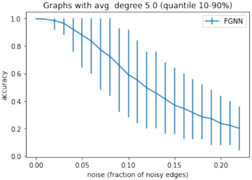
(c) Train - Regular, Test - ER



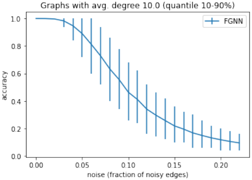
(d) Train - ER, Tested - Regular

Plan

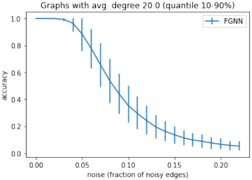
Different number of nodes



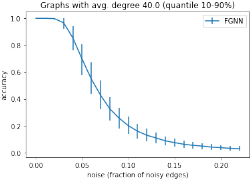
(e) $|V| = 25,\ d = 0.2$



(f) $|V| = 50,\ d = 0.2$

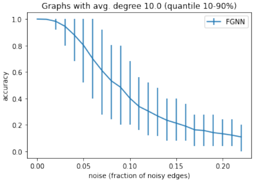


(g) $|V| = 100,\ d = 0.2$

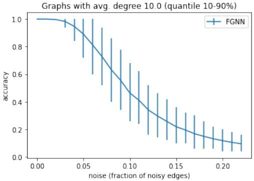


(h) $|V| = 200,\ d = 0.2$

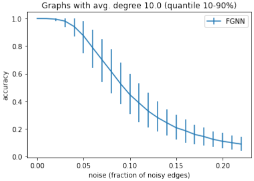
Same average degree



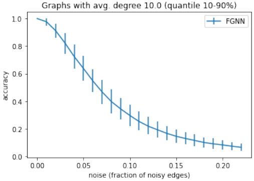
(i) $|V| = 25,\ d = 0.4$



(j) $|V| = 50,\ d = 0.2$



(k) $|V| = 100,\ d = 0.1$



(l) $|V| = 200,\ d = 0.05$

Plan

Conclusion

Résultats

1. Reproduction des résultats présentés dans le papier original ;
2. Unification de la phase d'apprentissage (sur GPU Google Colab) et de l'affichage des résultats sur un seul jupyter notebook ;
3. Étude des capacités de généralisation du modèle en termes de type de graphes et de nombre/densité de sommets.

Travaux futurs

Trouver une normalisation permettant une meilleure capacité de généralisation du réseau quant aux modifications du nombre de noeuds ou de la densité des arêtes.

Plan

References I

Waïss Azizian and Marc Lelarge, *Expressive power of invariant and equivariant graph neural networks*, International Conference on Learning Representations, 2021.