Stage M2/ingénieur
Apprentissage de distance d'éditions entre graphes par Réseaux de Neurones

Sébastien Bougleux[1], Luc Brun[1], Benoit Gaüzère[2], and Florian Yger[3]

[1]GREYC, ENSICAEN
[2]LITIS, INSA Rouen Normandie
[3]LAMSADE, Université Paris-Dauphine, PSL

25 janvier 2021

Description du problème à résoudre

La définition d'une notion de similarité ou de dissimilarité entre objets est souvent un préalable à toute définition d'une méthode d'apprentissage sur ces objets. Dans le domaine des graphes, la distance d'édition (*Graph Edit Distance* -GED-) constitue une mesure de dissimilarité privilégiée car facilement interprétable.

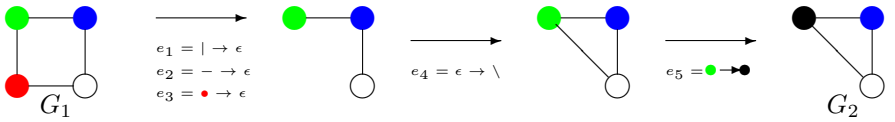

FIGURE 1 – Chemin d'édition $e_1, \ldots, e_5$ de G_1 à G_2.

Intuitivement, tout graphe peut être transformé en un autre graphe par une série d'opérations élémentaires (ajout/suppression d'un noeud ou d'une arête et changement du label d'un noeud ou d'une arête). Une telle séquence d'opérations est appelée un chemin d'édition (Figure 1). On associe également un coût à chaque opération élémentaire, le coût d'un chemin d'édition étant alors défini comme la somme des coûts de ses opérations élémentaires. Finalement, la distance d'édition entre deux graphes est définie comme le coût minimal du chemin d'édition permettant de transformer un graphe en un autre. On considère ainsi que deux graphes seront d'autant plus similaires que leur chemin d'édition minimal a un coût faible [1].

La distance d'édition peut donc se concevoir comme l'ampleur des modifications à apporter à un graphe pour le transformer en un autre. Toutefois, cette distance dépend des coûts fixés pour les opérations élémentaires. Bien définir

1. Naturellement, deux graphes isomorphes auront une distance d'édition nulle.

ces coûts est primordial pour bien capturer les dissimilarités entre graphes et donc inférer de bonnes propriétés à partir des graphes.

Généralement, ces coûts sont définis *a priori* par l'utilisateur, à partir d'une certaine expertise du problème à traiter et de certaines propriétés attendues. Cette stratégie implique une connaissance parfaite du domaine d'application et n'offre aucune garantie d'optimalité par rapport à la tâche d'apprentissage.

L'apprentissage de métrique a montré qu'il était en général plus efficace d'apprendre une notion de dissimilarité à partir des données plutôt que d'utiliser des dissimilarités naïves. Il en est de même pour les données structurées et plusieurs approches essayent d'apprendre les coûts associés à chaque opération d'édition. Ces approches sont probabilistes [6, 7], elles cherchent à maximiser la distance entre deux graphes ayant des propriétés différentes, et à minimiser la distance entre deux graphes ayant des propriétés similaires. Plus récemment, [1] a amélioré ce type d'approche en définissant le concept de "bonne" mesure de dissimilarité. Cependant, l'application de ces approches à des graphes génériques reste à étudier.

Ces dernières années ont vu l'essor des réseaux de neurones profonds. La force de ces modèles repose sur leur capacité à construire des représentations tirant partie des invariances propres aux données. Après avoir révolutionné le traitement d'image, ces modèles ont été naturellement étendus au domaine des graphes [2], avec des contributions majeures pour l'apprentissage de représentation de graphes [3, 10, 11].

L'objectif de ce stage est d'appliquer des réseaux de neurones pour l'apprentissage de coûts d'édition. En effet, utiliser une description optimisée des noeuds devrait permettre d'affiner la fonction de coût d'édition selon un objectif précis (classification, régression). Des travaux préliminaires, effectués au sein du GREYC, ont amené à des résultats encourageants qu'il convient d'approfondir. Ces travaux seront à mettre en relation avec ceux effectués dans [8] et [5]. La première mission de ce stage consistera en une étude et une évaluation approfondies de cette première méthode sur des jeux de données standards [4,9].

Déroulement du stage

Le stage est rémunéré et pourra se dérouler soit à Rouen (LITIS) ou à Caen (GREYC). En cas d'empêchement un stage à domicile sera également possible avec des points réguliers en visio conférence.

Le stage pourra se poursuivre en thèse selon les résultats obtenus et les financements disponibles. Plusieurs demandes de financement (ANR, thèse IA, thèse normale) sont actuellement en cours.

Profil du stagiaire

Les qualités attendues du stagiaire sont :

— Connaissance de Python et de Pytorch ou toute autre bibliothèque de réseaux de neurones,
— Capacités à lire et comprendre des articles scientifiques (en anglais),
— Autonomie dans le travail,

Contacts

Les personnes intéressées ou désirant plus d'informations doivent contacter :
— benoit.gauzere@insa-rouen.fr ;
— sebastien.bougleux@unicaen.fr ;
— luc.brun@e.email ;
— florian.yger@dauphine.fr.

Références

[1] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A Survey on Metric Learning for Feature Vectors and Structured Data. 2013.

[2] Martin Grohe. word2vec, node2vec, graph2vec, x2vec : Towards a theory of vector embeddings of structured data. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 1–16, 2020.

[3] William L. Hamilton. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3) :1–159, sep 2020.

[4] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark : Datasets for Machine Learning on Graphs. pages 1–33, 2020.

[5] Linlin Jia, Benoit Gaüzère, Florian Yger, and Paul Honeine. A metric learning approach to graph edit costs for regression. *Proceedings of S+SSPR*, 2021.

[6] Michel Neuhaus and Horst Bunke. A probabilistic approach to learning costs for graph edit distance. *Proceedings - International Conference on Pattern Recognition*, 3(C) :389–393, 2004.

[7] Michel Neuhaus and Horst Bunke. Automatic learning of cost functions for graph edit distance. *Information Sciences*, 177(1) :239–247, 2007.

[8] Pau Riba, Andreas Fischer, Josep Lladós, and Alicia Fornés. Learning Graph Edit Distance by Graph Neural Networks. 2020.

[9] Kaspar Riesen and Horst Bunke. IAM graph database repository for graph based pattern recognition and machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5342 LNCS :287–297, 2008.

[10] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1) :61–80, 2009.

[11] Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep Learning on Graphs : A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 14(8) :1–1, 2020.