S2-MINEUR - IA programmation

Chapitre 4 - Introduction aux Graph Neural Network (GNN)

MULAPI TITA Ketsia

Année académique 2024-2025

Les tâches

Figure 1: Graph classification/regression (in : a graph / out : a label)

Les tâches

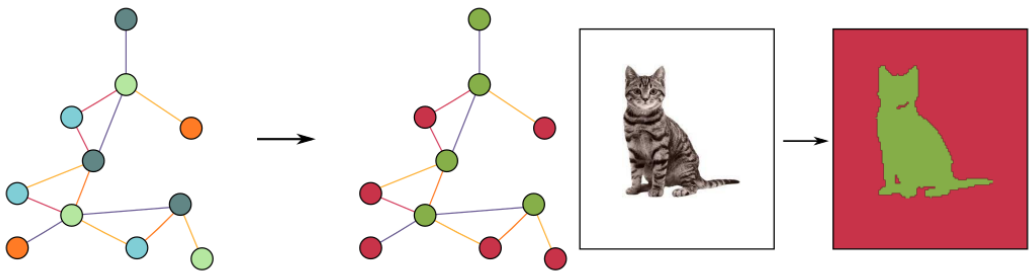


Figure 2: Nodes classification/regression (in : a graph / out : a labelled graph)

Les tâches

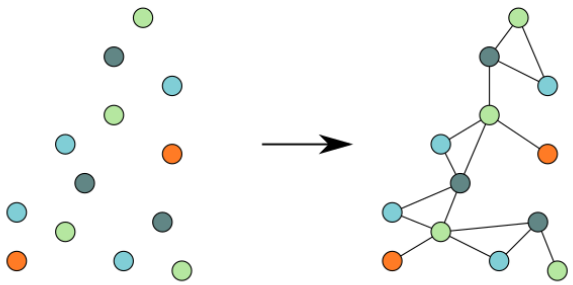


Figure 3: Link prediction (in : pair of nodes / out : existence of an edge)

Les tâches

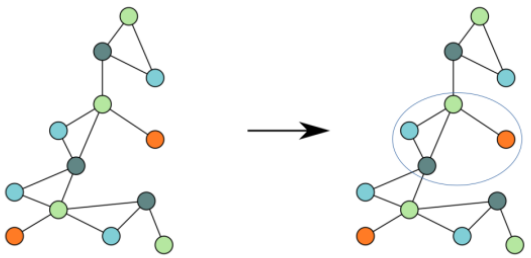


Figure 4: Community detection (in : a graph / out : a subgraph)

Les tâches

A stop sign is flying in blue skies.

John is in the kitchen...
...The cat is near John

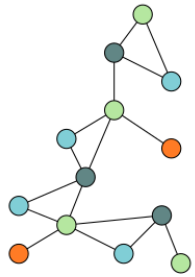


Figure 5: text-image to graph

Les tâches

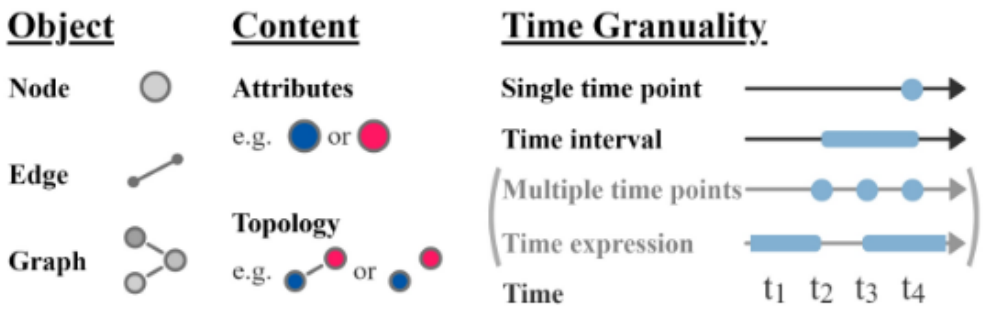


Figure 6: anything to graph

Graph Pooling

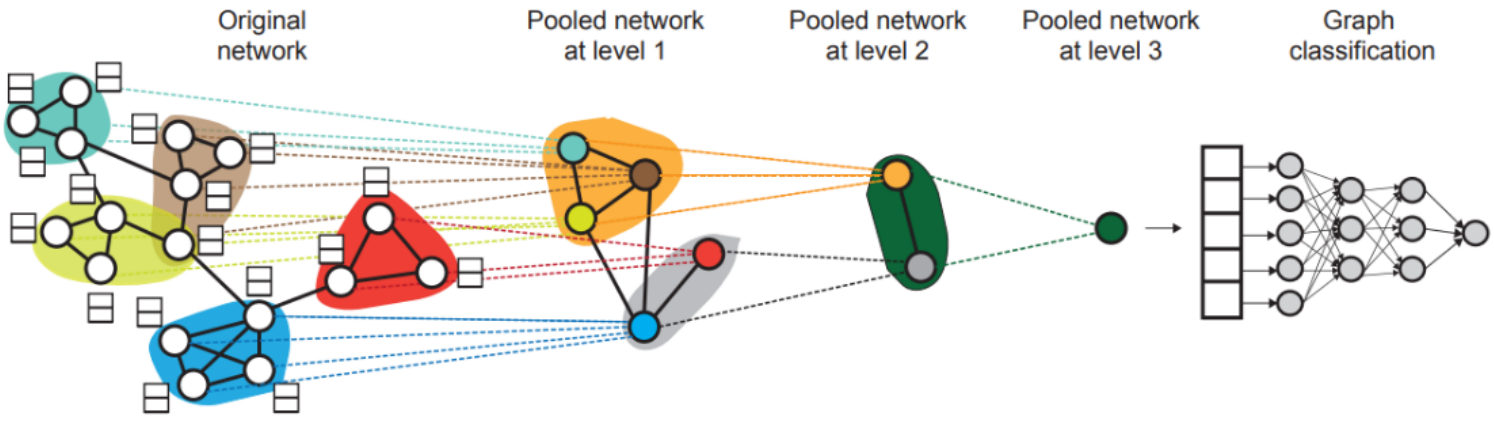


Figure 7: ying2018hierarchical (hierarchical pooling method)

Graph Similarity

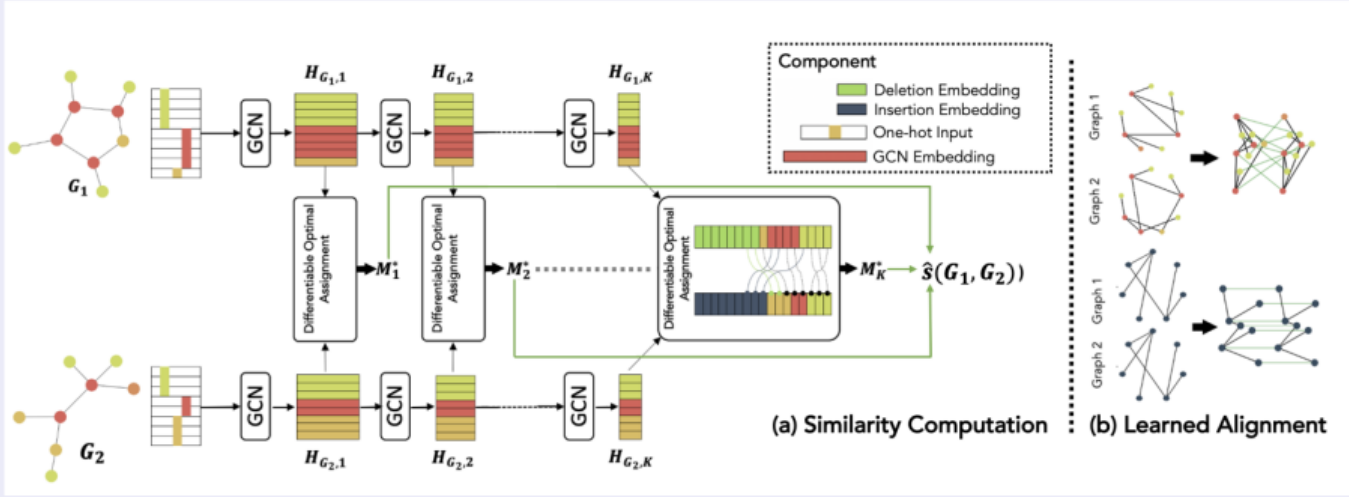


Figure 8: Graphes Siamois

A word about reference datasets

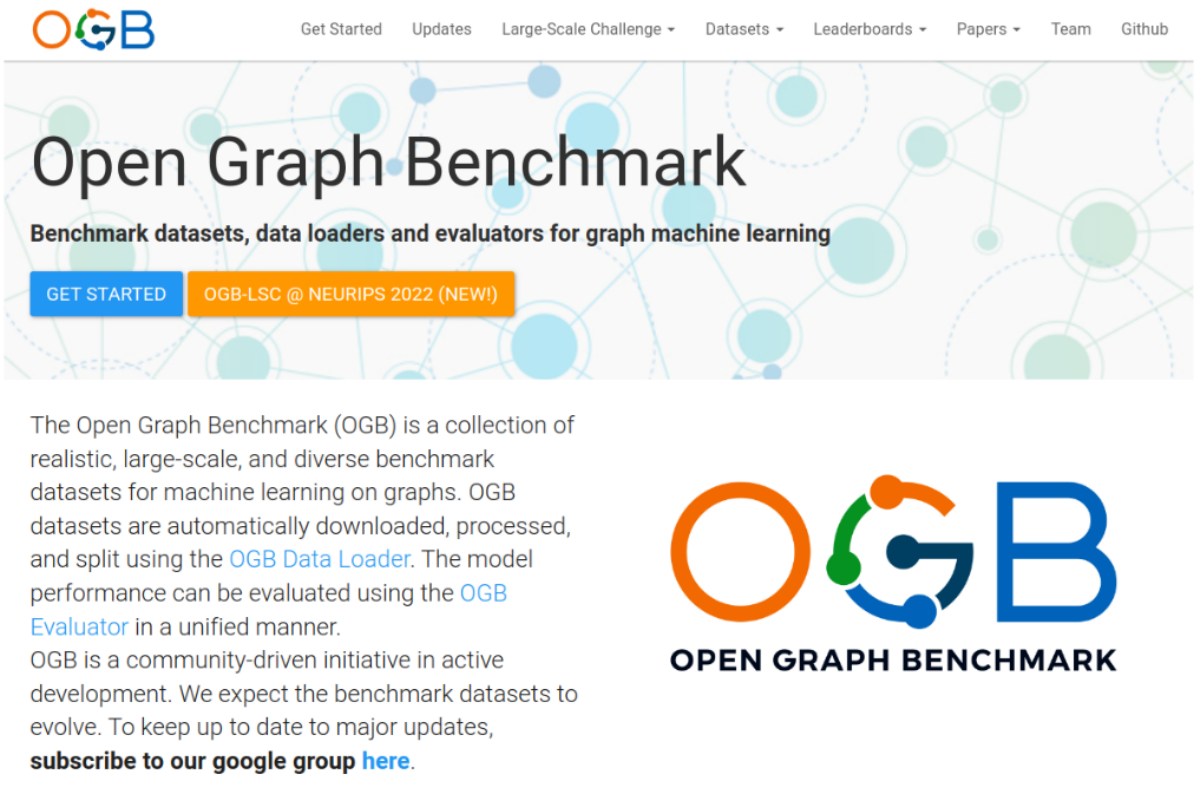
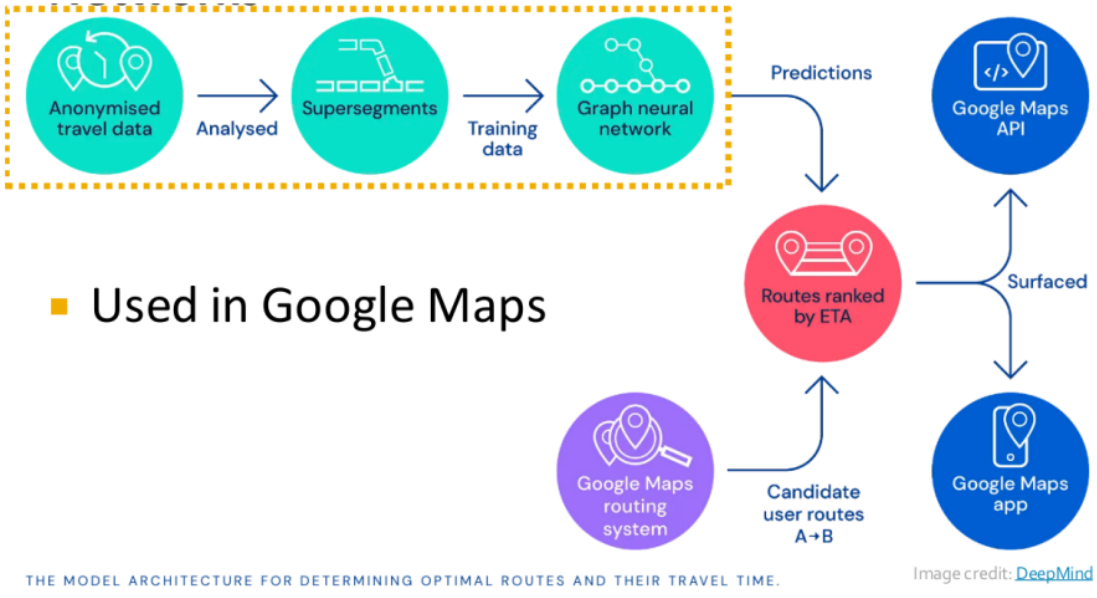


Figure 9: Enter Caption

Few recent application examples

At the subgraph level : estimated time of arrival (ETA)

- Nodes = Road segments
- Edges = Connectivity between road segments
- Machine learning task : predict the time of arrival
- Key idea : Use of a graph neural network



- Used in Google Maps

Figure 10: A découvrir !

4.1. Introduciton au ML non supervisé sur les graphes

Qu'est-ce que l'apprentissage non supervisé ?

L'apprentissage non supervisé consiste à apprendre sans labels :

- On ne sait pas à l'avance ce qu'on recherche précisément
- Le but est de découvrir automatiquement des motifs, des regroupements ou des anomalies

Exemple concret : Identifier automatiquement des communautés d'amis sur Facebook sans connaître leurs intérêts.

Pourquoi utiliser l'embedding non supervisé pour les graphes ?

Les graphes sont complexes et difficiles à analyser directement :

- Ils ne respectent pas la géométrie classique (non-euclidienne)
- Besoin d'une représentation numérique simplifiée : embedding

Exemple concret : Représenter chaque utilisateur Twitter par un vecteur, rapprochant ceux qui suivent les mêmes comptes.

Méthodes simples : les Shallow Embeddings

Ces méthodes sont intuitives et rapides à calculer :

- Utilisent des parcours aléatoires ou la décomposition matricielle
- Capturent bien les structures globales et locales

Exemples pratiques :

- DeepWalk : Parcours aléatoires aléatoires simples
- Node2Vec : Parcours aléatoires avec biais, plus précis

La factorisation de matrices

Idée intuitive :

- Décomposer une grande matrice (Adjacence : relations entre les nœuds) en deux matrices plus simples (U la matrice d'embedding (ou de caractéristiques latentes des nœuds) et V, la matrice de contexte (ou d'interaction entre ces caractéristiques))
- Chaque matrice simple décrit les nœuds avec des caractéristiques cachées

Exemple : identifier implicitement des groupes dans un réseau professionnel (LinkedIn).

Voir aussi des approches comme SDNE, Graph Factorization (GF) ou Higher-Order Proximity Embedding (HOPE).

Modèles Skip-Gram appliqués aux graphes

Ces modèles viennent de la linguistique :

Skip-Gram de Word2Vec est une approche où chaque nœud central (comme un mot) prédit ses voisins de contexte (nœuds proches dans la séquence aléatoire), autrement, **DeepWalk transforme un graphe en phrases via des balades aléatoires, puis applique Word2Vec pour apprendre des vecteurs de nœuds.** Deux nœuds proches dans les trajets aléatoires auront des embeddings similaires.Chaque parcours aléatoire devient une phrase, chaque nœud un motLe modèle apprend à prédire les voisins (contextes fréquents)

Exemples pratiques :

- DeepWalk : simple et efficace (C'est une méthode d'embedding de graphe qui transforme un graphe en phrases de mots, puis utilise Word2Vec pour apprendre les vecteurs des nœuds)
- Node2Vec : captures nuances dans les structures locales/globales

Autoencodeurs

Un autoencodeur est comme un compresseur intelligent :

- Il apprend à reproduire l'information initiale après l'avoir compressée
- **La compression révèle des caractéristiques cachées et importantes**

Exemples : Compression d'images, réduction de bruit audio.

Autoencodeurs pour les graphes

Adaptation des autoencodeurs aux graphes :

- Objectif : reconstruire la matrice d'adjacence (relations)
- Capturent les proximités directes et indirectes

Exemple concret : Trouver des utilisateurs similaires sur Instagram grâce à leurs connexions indirectes.

Graph Neural Networks (GNNs) : explication intuitive

Les GNNs sont des réseaux de neurones adaptés aux graphes :

- Chaque nœud agrège des informations venant de ses voisins directs
- Permet d'apprendre des structures complexes

Exemple simple : Identifier des communautés d'intérêt dans les forums en ligne.

L'approche spectrale

En GNN, l'approche spectrale Utilise la théorie spectrale des graphes (décomposition du Laplacien) car elle est basée sur une transformation mathématique (domaine fréquentiel)

Par exemple, la convolution spectrale traite le graphe comme une musique (analogie musicale) Intuition :

- Imaginez le graphe comme une musique décomposée en différentes fréquences (spectre)
- Chaque fréquence capture un type de relation dans le graphe

Exemple : Graph Convolutional Network (GCN) pour détecter des groupes d'intérêts similaires dans un réseau professionnel.

L'approche spatiale

Et puis toujours dnas les GNNS, on a l'approche spatiale qui permet de travailler directement dans l'espace du graphe (intuitive et scalable), c'est-à-dire, avec les voisins dans du domaine tout en agrégant localement les informations sans passer par une transformation mathématique complexe :

- Chaque nœud reçoit directement des informations de ses voisins
- Plus facile à calculer pour de grands réseaux

Exemple : GraphSAGE pour recommander automatiquement de nouveaux contacts LinkedIn (en analysant rapidement les relations directes et indirectes).

Résumé des points essentiels

Points clés à retenir :

- L'apprentissage non supervisé aide à comprendre les graphes sans labels
- Trois grandes approches : shallow embeddings, autoencodeurs, GNNs (Specatrale, Spatiale).
- Choix dépend de la complexité du problème, de la taille du graphe, du type de graphe et des objectifs d'analyse

Étape suivante : utiliser ces embeddings pour des prédictions ou pour identifier des anomalies.

4.2. Machine Supervised Learning on Graphs

Introduction à l'apprentissage supervisé sur graphes

L'apprentissage supervisé utilise des données étiquetées pour apprendre une fonction de prédiction :

- Les données comprennent des couples (entrée, étiquette)
- Objectif : prédire correctement les étiquettes des nouvelles données
- Exemple : identifier les utilisateurs d'un réseau social susceptibles de fermer leur compte

Feuille de route de l'embedding supervisé

Principales approches en apprentissage supervisé sur graphes :

- Méthodes basées sur les caractéristiques (features)
- Méthodes d'embedding superficiel (shallow embeddings)
- Méthodes de régularisation par graphe
- Réseaux neuronaux convolutifs sur graphes (Graph CNN ou GCN)

Méthodes basées sur les caractéristiques (features)

Approche directe utilisant des propriétés descriptives du graphe :

1. Sélection de propriétés significatives (ex : nombre de liens, clustering moyen, efficacité globale)
2. Utilisation de ces propriétés dans des algorithmes classiques (SVM, arbres décisionnels)

Exemple concret : classification de protéines en enzymes ou non-enzymes.

Shallow Embedding : principe et usage

Méthodes simples mais efficaces, appliquées uniquement aux données d'entraînement :

- Exemple : propagation d'étiquettes (Label Propagation)
- Les étiquettes des nœuds connus influencent les nœuds inconnus par proximité

Application pratique : identifier des groupes communautaires sur les réseaux sociaux.

Propagation d'étiquettes en détail

Principe mathématique :

- Construction d'une matrice de transition basée sur l'adjacence
- Itérations jusqu'à convergence pour diffuser les étiquettes

Résultat concret : assignation probabiliste des étiquettes aux nœuds inconnus.

Label Spreading : amélioration de la propagation d'étiquettes

Résout les limites de Label Propagation en autorisant les modifications d'étiquettes initiales :

- Utilisation du Laplacien normalisé pour contrôler la diffusion
- Pondération de l'influence des étiquettes initiales

Avantage concret : robustesse accrue face aux erreurs d'étiquetage initiales.

Méthodes de régularisation par graphe

Utilisation de l'information topologique du graphe pour régulariser l'apprentissage :

- Contraint la fonction prédictive à être lisse entre voisins
- Applicable aux réseaux neuronaux pour prévenir le sur-apprentissage

Exemple : régularisation Laplacienne.

Apprentissage par régularisation de variétés

Extension des méthodes précédentes en imposant la régularité sur l'espace de représentation intermédiaire (embedding) :

- Application à différentes couches intermédiaires du réseau
- Contrôle fin de l'influence topologique sur l'apprentissage

Neural Graph Learning (NGL)

Généralisation des approches précédentes dans le cadre des réseaux neuronaux :

- Combinaison flexible d'informations étiquetées et non-étiquetées
- Intégration directe dans les frameworks comme TensorFlow

Exemple pratique : classification des documents scientifiques (dataset Cora).

Graph CNN : convolutions sur graphes

Réseaux neuronaux adaptés à la structure non-euclidienne des graphes :

- Convolutions spectrales : utilisent la décomposition en fréquences du graphe
- Convolutions spatiales : agrègent directement les informations des voisins

Application concrète : classification des graphes de protéines.

Résumé des points clés

Points essentiels à retenir :

- L'apprentissage supervisé exploite les étiquettes pour prédire efficacement
- Ici on a 4 méthodes diversifiées : caractéristiques, shallow embeddings, régularisation, Graph CNN
- Choisir selon la structure des données, objectifs et taille du graphe

Applications pratiques : classification de nœuds, prédiction d'attributs (ex. moléculaires), et régularisation robuste.

GCN vs GAT

4.2.4 GAT (Graph Attention Networks)

Les réseaux à attention pour graphes (GAT) introduisent un mécanisme d'attention pour pondérer les voisins d'un nœud de façon adaptative :

- Chaque nœud attribue des poids d'attention à ses voisins directs.
- L'attention permet au modèle de focaliser sur les voisins les plus pertinents.
- Flexibilité accrue dans l'apprentissage des représentations des graphes.

Illustration pratique :

- Identifier les utilisateurs influents sur Twitter en fonction de l'attention portée à leurs interactions les plus significatives.

4.2.5 GCN vs GAT : expressivité, robustesse, scalabilité

Expressivité :

- **GCN** : Capturent efficacement les structures locales mais avec des pondérations fixes.
- **GAT** : Plus expressif grâce aux pondérations adaptatives via le mécanisme d'attention.

Robustesse :

- **GCN** : Robuste mais peut être limité par des hypothèses implicites sur les voisinages.
- **GAT** : Potentiellement plus sensible au bruit, mais généralement plus flexible et performant.

Scalabilité :

- **GCN** : Plus simple à calculer et facilement scalable pour des grands graphes.
- **GAT** : Nécessite des calculs d'attention additionnels, pouvant poser des défis en scalabilité sur très grands graphes.

Travaux pratiques

1. Vous allez accéder au dossier Introduction aux GNN, présent sur moodle, il contient 3 fichiers pdf et 2 dossier :
   1.1 01_Graph-embeddings.ipynb
   1.2 02_prediction_des_liens.ipynb
   1.3 03_GCNN.ipynb
   1.4 cora/
   1.5 nxt_gem/
2. Cette séance est une séance d'introduction, je vous encourage à aller plus loin en faisant par exemple du GNN-NLP, etc.
3. Si vous avez besoin de plus de documentation, contactez-moi ! contact@ketiaamulapi.com