Graph Analytics et apprentissage sur graphes

Méthodes structurelles, spectrales et réseaux de neurones graphiques

Redha Moulla

Plan

PARTIE I

Fondements, représentations et tâches

Objets et relations

Un graphe attribué est décrit par

$$G = (V, E, X, E_f), \qquad |V| = n, \quad |E| = m,$$

où $X \in \mathbb{R}^{n \times d}$ contient les attributs des nœuds et E_f ceux des arêtes.

Structure

- V : entités, positions, documents, molécules ;
- $E \subseteq V \times V$: relations observées ; pour un graphe non orienté, on peut employer des paires non ordonnées ;
- orientation, poids, type et temps peuvent enrichir une arête.

Signal

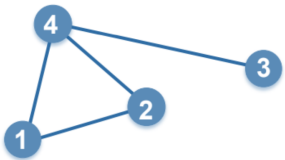
- attributs continus ou catégoriels ;
- labels partiels sur les nœuds, arêtes ou graphes ;
- variables cibles dépendant simultanément du signal et de la topologie.

Niveaux de prédiction

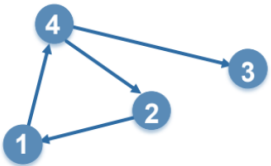
Niveau	**Variable cible**	**Exemples**	**Sortie**
Nœud	y_v	fraude d'un compte, rôle d'une protéine	n prédictions
Arête / paire	y_{uv}	lien futur, compatibilité, transaction suspecte	score $s(u, v)$
Graphe	y_G	propriété moléculaire, classe d'un programme	une prédiction par graphe
Sous-graphe	y_S	motif fonctionnel, communauté, événement	ensemble ou masque

Le choix de la cible impose la représentation finale : embedding de nœud, fonction de paire ou agrégation globale.

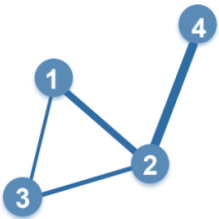
Types de graphes



Non orienté
$A = A^\top$



Orienté
$A_{ij} \neq A_{ji}$ en général



Pondéré
$A_{ij} = w_{ij}$

Un graphe peut être simultanément orienté, pondéré, multirelationnel et dynamique. Ces propriétés ne doivent pas être supprimées lors du prétraitement si elles portent du signal.

Matrice d'adjacence et listes d'adjacence

Pour un graphe simple non orienté,

$$A_{ij} = \mathbf{1}[\{i,j\} \in E], \qquad A = A^\top.$$

Matrice dense

- accès à une paire en $O(1)$;
- stockage $O(n^2)$;
- pertinente pour des graphes denses ou de petite taille.

Structure creuse

- listes d'adjacence ou formats COO/CSR ;
- stockage $O(n+m)$;
- produit AX en $O(md)$, sans construire une matrice dense.

Degrés et voisinages

Pour un graphe non orienté,

$$d_i = \sum_j A_{ij}, \qquad D = \mathrm{diag}(d_1, \ldots, d_n), \qquad \mathcal{N}(i) = \{j : A_{ij} \neq 0\}.$$

Dans un graphe orienté, on distingue

$$d_i^{\mathrm{out}} = \sum_j A_{ij}, \qquad d_i^{\mathrm{in}} = \sum_j A_{ji}.$$

- Le voisinage à k sauts est porté par les puissances de A.
- Le nombre de marches de longueur k de i à j est $(A^k)_{ij}$.
- Un degré élevé ne signifie ni centralité globale ni causalité.

Chemins, marches et distances

- Une **marche** autorise la répétition de sommets et d'arêtes.
- Un **chemin simple** ne répète pas de sommet.
- La distance géodésique $d(i, j)$ est la longueur d'un plus court chemin.
- Dans un graphe pondéré, la longueur est la somme des coûts d'arêtes.

Algorithmes

Poids	Plus courts chemins
unitaires	BFS, $O(n + m)$
non négatifs	Dijkstra
poids négatifs	Bellman–Ford
toutes paires	Floyd–Warshall

Matrice d'incidence

Après avoir orienté arbitrairement chaque arête, la matrice d'incidence $B \in \mathbb{R}^{n \times m}$ est définie par

$$B_{ve} = \begin{cases} +1 & \text{si } v \text{ est la tête de } e, \\ -1 & \text{si } v \text{ est la queue de } e, \\ 0 & \text{sinon.} \end{cases}$$

Pour un graphe non orienté, cette orientation ne change pas

$$L = BB^\top = D - A.$$

- $B^\top x$ calcule les différences du signal x le long des arêtes.
- $x^\top L x = \|B^\top x\|_2^2 = \sum_{\{i,j\} \in E} (x_i - x_j)^2$.

Attributs du graphe

Support	Exemples	Traitement
Nœud x_v	catégorie, texte, coordonnées, mesures	normalisation, encodage, modèle multimodal
Arête e_{uv}	montant, distance, type, horodatage	message conditionné par l'arête
Graphe g	contexte, température, domaine	concaténation au readout ou conditionnement
Structure	degré, motifs, positions spectrales	encodages structurels calculés sur le graphe

Deux nœuds de même degré peuvent jouer des rôles très différents ; deux nœuds aux attributs proches peuvent être séparés par la structure.

Cadres transductif et inductif

Transductif

Le graphe complet est connu pendant l'entraînement ; seuls certains labels sont masqués.

- les nœuds de test participent à la propagation ;
- la généralisation porte sur des labels inconnus du même graphe.

Inductif

Les nœuds ou graphes de test sont absents de l'entraînement.

- le modèle doit produire une représentation pour un nouvel objet ;
- l'évaluation mesure la généralisation hors du graphe observé.

Source : Hamilton, Ying et Leskovec, 2017.

Permutation des nœuds

Soit P une matrice de permutation. Un renommage des nœuds transforme

$$A' = PAP^\top, \qquad X' = PX.$$

Un modèle de nœuds doit être **équivariant** :

$$f(PAP^\top, PX) = Pf(A, X).$$

Un modèle de graphe doit être **invariant** :

$$g(PAP^\top, PX) = g(A, X).$$

Les agrégations symétriques — somme, moyenne, maximum — respectent cette contrainte ; une concaténation dépendant de l'ordre des voisins ne la respecte pas.

PARTIE II

Mesures structurelles et communautés

Mesures de centralité

Mesure	Question	Hypothèse implicite
Degré	combien de voisins directs ?	l'activité locale suffit
Proximité	à quelle distance des autres ?	les chemins courts portent l'accès
Intermédiarité	combien de géodésiques traversent le nœud ?	le flux suit les plus courts chemins
Vecteur propre	suis-je relié à des nœuds eux-mêmes importants ?	importance récursive
PageRank	quelle masse stationnaire reçoit le nœud ?	navigation aléatoire avec téléportation

Le classement dépend donc du mécanisme étudié ; il n'existe pas de centralité universelle.

Degré, force et coefficient de clustering

Pour un graphe non orienté,

$$d_i = \sum_j A_{ij}, \qquad s_i = \sum_j w_{ij}.$$

Le coefficient de clustering local mesure la fermeture des triangles :

$$C_i = \frac{2T_i}{d_i(d_i - 1)},$$

où T_i est le nombre d'arêtes entre voisins de i. On pose $C_i = 0$ par convention lorsque $d_i < 2$.

- d_i mesure la connectivité locale ; s_i tient compte de l'intensité des liens.
- C_i distingue un hub reliant des voisins séparés d'un nœud inséré dans un groupe dense.

Centralité de proximité

Dans un graphe connexe,

$$C_C(i) = \frac{n-1}{\sum_{j \neq i} d(i,j)}.$$

Pour un graphe non connexe, la centralité harmonique reste définie :

$$C_H(i) = \sum_{j \neq i} \frac{1}{d(i,j)}, \qquad \frac{1}{\infty} = 0.$$

Interprétation

- accès rapide au reste du réseau ;
- mesure globale sensible aux raccourcis.

Calcul

- BFS répété pour un graphe non pondéré ;
- Dijkstra répété pour des coûts non négatifs.

Centralité d'intermédiarité

Soit σ_{st} le nombre de plus courts chemins entre s et t, et $\sigma_{st}(v)$ ceux qui passent par v :

$$C_B(v) = \sum_{\substack{s,t \in V \setminus \{v\} \\ s \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

Pour une arête e,

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}.$$

Dans un graphe non orienté, on somme sur les paires $\{s, t\}$ non ordonnées, ou l'on divise par deux la somme sur les couples ordonnés. Une normalisation par le nombre de paires est souvent ajoutée.

- Une forte intermédiarité signale un pont entre régions du graphe.
- La suppression successive des arêtes de forte intermédiarité fonde l'algorithme de Girvan–Newman.
- Le calcul exact peut être coûteux ; l'algorithme de Brandes l'effectue en $O(nm)$ sur un graphe non pondéré.

Centralité de vecteur propre

La centralité x_i est proportionnelle à la centralité des voisins :

$$x_i = \frac{1}{\lambda} \sum_j A_{ij} x_j \quad \Longleftrightarrow \quad Ax = \lambda x.$$

Sous les hypothèses de Perron–Frobenius, le vecteur propre dominant peut être choisi positif.

- La mesure privilégie les nœuds connectés à des nœuds déjà centraux.
- Elle peut se concentrer sur une composante dominante ou sur un sous-graphe très dense.
- La méthode de la puissance calcule $x \leftarrow Ax/\|Ax\|$.

PageRank

On remplace d'abord chaque colonne pendante, correspondant à un sommet sans sortie, par une distribution de probabilité. Pour la matrice colonne-stochastique P ainsi obtenue, PageRank cherche

$$r = \alpha P r + (1 - \alpha) v,$$

où v est une distribution de téléportation et $0 < \alpha < 1$.

Solution

$$r = (1 - \alpha)(I - \alpha P)^{-1} v.$$

Si v est strictement positif, la matrice de Google $\alpha P + (1 - \alpha) v \mathbf{1}^\top$ est positive : elle admet une unique distribution stationnaire. Le vecteur v permet aussi une personnalisation du classement.

Source : Brin et Page, 1998.

Prédiction locale de liens

Pour deux nœuds non reliés u, v, les scores classiques sont

$$\mathrm{CN}(u,v) = |\mathcal{N}(u) \cap \mathcal{N}(v)|,$$

$$\mathrm{Jaccard}(u,v) = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v)|},$$

$$\mathrm{AA}(u,v) = \sum_{z \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{\log d_z}.$$

Adamic–Adar réduit la contribution des voisins très fréquents ; Jaccard corrige partiellement l'effet du degré.

Indice de Katz

Le score de Katz est

$$S = \sum_{\ell=1}^{\infty} \beta^{\ell} A^{\ell} = (I - \beta A)^{-1} - I, \qquad 0 < \beta < \rho(A)^{-1}.$$

Ainsi,

$$S_{uv} = \sum_{\ell \geq 1} \beta^{\ell} (A^{\ell})_{uv}.$$

- Les chemins courts reçoivent un poids supérieur.
- Le calcul exact est global ; des solveurs linéaires ou des troncatures sont nécessaires à grande échelle.
- Le score peut favoriser les zones de fort degré sans normalisation adaptée.

Modularité

Pour une partition c_i des nœuds d'un graphe non orienté,

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \mathbf{1}[c_i = c_j].$$

Le terme $d_i d_j/(2m)$ correspond au nombre attendu d'arêtes dans le modèle de configuration.

- $Q > 0$ indique davantage d'arêtes internes que sous le modèle nul.
- La maximisation exacte est difficile ; Louvain et Leiden utilisent des heuristiques multi-échelles.
- La modularité possède une limite de résolution : de petites communautés peuvent être fusionnées.

Algorithme de Louvain

1. initialiser une communauté par nœud ;
2. déplacer chaque nœud vers la communauté voisine donnant le meilleur gain $\Delta Q > 0$;
3. contracter chaque communauté en un super-nœud pondéré ;
4. répéter jusqu'à stabilisation de la modularité.

Atout
Traitement rapide de grands graphes creux.

Conséquence
La solution dépend de l'ordre des mises à jour et peut varier entre exécutions.

Évaluation des communautés

Situation	Mesures	Ce qu'elles testent
Partition de référence	ARI, NMI, variation d'information	accord entre deux partitions
Pas de référence	modularité, conductance, densité interne	cohésion selon un critère structurel
Communautés chevauchantes	Omega index, F-score par ensemble	recouvrement partiel
Stabilité	consensus, variation sous rééchantillonnage	robustesse de la partition

Une forte modularité ne prouve pas que les groupes correspondent à une catégorie métier ; elle valide un contraste topologique relatif au modèle nul choisi.

PARTIE III

Méthodes spectrales et semi-supervisées

Laplaciens d'un graphe

Pour un graphe non orienté à degrés strictement positifs,

$$L = D - A, \qquad L_{\mathrm{rw}} = I - D^{-1}A, \qquad L_{\mathrm{sym}} = I - D^{-1/2}AD^{-1/2}.$$

Opérateur	**Propriété**	**Usage**
L	symétrique, dépend de l'échelle des degrés	énergie de Dirichlet, graphes réguliers
L_{rw}	lié à la marche $D^{-1}A$	diffusion, probabilités de transition
L_{sym}	symétrique et normalisé	clustering spectral, filtres spectraux

Énergie de Dirichlet

Pour un signal $x \in \mathbb{R}^n$,

$$x^\top L x = \frac{1}{2} \sum_{i,j} A_{ij}(x_i - x_j)^2.$$

Donc $L \succeq 0$ et

$$L\mathbf{1} = 0.$$

- Une faible énergie correspond à un signal lisse sur les arêtes.
- La multiplicité de la valeur propre 0 est le nombre de composantes connexes.
- Si le graphe est connexe, le vecteur de Fiedler associé à $\lambda_2 > 0$ fournit une direction de séparation. Sinon, la multiplicité de 0 décrit d'abord les composantes connexes.

Transformée de Fourier sur graphe

Comme $L = U\Lambda U^\top$, tout signal s'écrit

$$x = U\widehat{x}, \qquad \widehat{x} = U^\top x.$$

Les petites valeurs propres correspondent à des variations lentes sur le graphe ; les grandes à des variations rapides.

Filtre spectral

$$g_\theta(L)x = U g_\theta(\Lambda) U^\top x.$$

Cette définition est globale et coûteuse si elle nécessite une décomposition spectrale complète.

Clustering spectral

Pour une partition $S, \bar{S}$, la coupe normalisée est

$$\mathrm{Ncut}(S, \bar{S}) = \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S)} + \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(\bar{S})}.$$

La relaxation continue conduit au problème

$$\min_{f \perp D^{1/2}\mathbf{1}} \frac{f^\top L_{\mathrm{sym}} f}{f^\top f},$$

dont la solution est le deuxième vecteur propre de L_{sym}.

- L'équilibrage par les volumes évite les coupes isolant un seul nœud.
- Pour $k > 2$, l'algorithme normalisé usuel emploie les k vecteurs propres associés aux plus petites valeurs propres, puis normalise les lignes de leur matrice.

Clustering spectral normalisé

1. construire A, D et $L_{\text{sym}} = I - D^{-1/2} A D^{-1/2}$;
2. extraire les k vecteurs propres associés aux plus petites valeurs propres ;
3. former $U \in \mathbb{R}^{n \times k}$ et normaliser ses lignes ;
4. appliquer k-means aux lignes normalisées ;
5. reporter les clusters obtenus sur les nœuds.

Le coût dominant est l'extraction de sous-espace propre ; pour un graphe creux, Lanczos évite une diagonalisation dense.

Choix du nombre de clusters

Eigengap

Choisir k lorsque

$$\lambda_{k+1} - \lambda_k$$

est nettement plus grand que les écarts voisins.

Stabilité

Comparer les partitions obtenues sous

- perturbation des arêtes ;
- variations d'hyperparamètres ;
- initialisations de k-means.

L'eigengap est un indice spectral, pas une garantie métier sur le nombre de groupes.

Propagation de labels

On sépare les nœuds étiquetés L et non étiquetés U. Pour des scores $F \in \mathbb{R}^{n \times c}$, on minimise

$$\frac{1}{2} \sum_{i,j} A_{ij} \, \|F_i - F_j\|_2^2 = \operatorname{tr}(F^\top L F)$$

sous la contrainte $F_L = Y_L$.

Solution harmonique

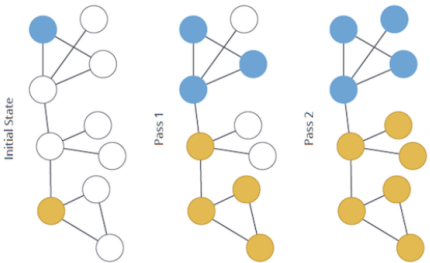
$$F_U = -L_{UU}^{-1} L_{UL} Y_L.$$

Cette solution est unique si chaque composante connexe contenant un nœud non étiqueté contient aussi au moins un nœud étiqueté ; sinon L_{UU} est singulière sur une composante sans label. La méthode suppose en outre que les labels varient peu le long des arêtes.

Diffusion itérative des labels

Avec $S = D^{-1/2} A D^{-1/2}$, une mise à jour régularisée s'écrit

$$F^{(t+1)} = \alpha S F^{(t)} + (1 - \alpha) Y, \qquad 0 < \alpha < 1.$$



- le terme $SF^{(t)}$ diffuse les scores ;
- le terme $(1 - \alpha)Y$ maintient l'information initiale ;
- la limite résout un système linéaire :

$$F^* = (1 - \alpha)(I - \alpha S)^{-1} Y.$$

Limites du lissage sur graphe

- **Hétérophilie :** les voisins portent souvent des labels différents ; lisser détruit le signal.
- **Composantes sans label :** aucune information supervisée ne peut y être propagée.
- **Hub dominant :** un nœud de très fort degré peut diffuser un biais à grande échelle.
- **Labels bruités :** une erreur initiale se propage aux voisins.
- **Fuite de protocole :** construire le graphe avec des données futures ou des liens de test contamine l'évaluation.

Le choix de l'opérateur de diffusion doit être cohérent avec le mécanisme supposé de corrélation sur le graphe.

PARTIE IV

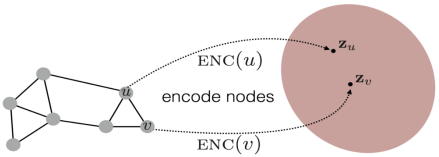
Embeddings de nœuds

Embeddings de nœuds

On apprend une application

$$z : V \longrightarrow \mathbb{R}^d, \qquad d \ll n,$$

afin que la géométrie de l'espace latent conserve une notion de proximité du graphe.



- proximité de premier ordre : arêtes observées ;
- proximité de second ordre : voisinages similaires ;
- équivalence structurelle : rôles analogues sans proximité géodésique ;
- objectif supervisé : géométrie adaptée à une cible.

Proximités préservées

Famille	**Signal préservé**	**Exemples**
Factorisation	matrice de proximité explicite	décomposition spectrale, HOPE
Marches aléatoires	cooccurrences dans des contextes	DeepWalk, node2vec
Autoencodeur	reconstruction des arêtes ou attributs	graph autoencoder, VGAE
Message passing	fonction paramétrique des attributs et voisins	GCN, GraphSAGE, GAT

Deux méthodes ne sont comparables que si leur notion de voisinage et leur protocole d'évaluation sont compatibles.

Marches aléatoires et corpus

Une marche $v_0, v_1, \ldots, v_T$ suit

$$\Pr(v_{t+1} = x \mid v_t = v) = \frac{w_{vx}}{\sum_{u \in \mathcal{N}(v)} w_{vu}}.$$

Chaque séquence devient une « phrase » ; les nœuds dans une fenêtre de contexte deviennent des paires positives.

- des marches courtes capturent le voisinage local ;
- leur répétition approxime des statistiques de diffusion ;
- le choix de la transition détermine le biais structurel du corpus.

DeepWalk

Pour chaque nœud de départ :

1. générer plusieurs marches aléatoires uniformes ;
2. extraire les paires (v, u) situées dans une fenêtre de rayon c ;
3. maximiser la vraisemblance des contextes observés ;
4. utiliser les vecteurs appris dans une tâche aval.

Objectif Skip-gram exact

$$\max_{Z,Z'} \sum_{(v,u)\in\mathcal{C}} \log \frac{\exp(z_v^\top z'_u)}{\sum_{w\in V} \exp(z_v^\top z'_w)}.$$

Source : Perozzi, Al-Rfou et Skiena, 2014.

Échantillonnage négatif

Pour une paire positive (v, u), on minimise

$$\ell_{v,u} = -\log \sigma(z_v^\top z'_u) - \sum_{r=1}^{K} \mathbb{E}_{w \sim P_n} \log \sigma(-z_v^\top z'_w).$$

- K contrôle le nombre de négatifs par paire positive.
- P_n est souvent une distribution lissée de fréquence.
- Les négatifs faux — arêtes non observées mais plausibles — introduisent un biais dans les graphes incomplets.

node2vec

Après la transition $t \to v$, le poids non normalisé d'un candidat $x \in \mathcal{N}(v)$ est

$$\pi_{vx} = \alpha_{pq}(t,x) w_{vx}, \qquad \alpha_{pq}(t,x) = \begin{cases} 1/p & d(t,x) = 0, \\ 1 & d(t,x) = 1, \\ 1/q & d(t,x) = 2. \end{cases}$$

Puis

$$\Pr(v_{t+1} = x \mid v_t = v, v_{t-1} = t) = \frac{\pi_{vx}}{\sum_{y \in \mathcal{N}(v)} \pi_{vy}}.$$

Source : Grover et Leskovec, 2016.

Paramètres p et q

Paramètre de retour p

- petit p : retour fréquent vers le nœud précédent ;
- grand p : décourage le demi-tour.

Paramètre d'exploration q

- $q > 1$: marche plus locale, proche d'une exploration BFS ;
- $q < 1$: exploration extérieure, proche d'une DFS.

Le réglage oppose davantage proximité communautaire et similarité de rôle qu'une séparation stricte BFS/DFS.

Interprétation matricielle

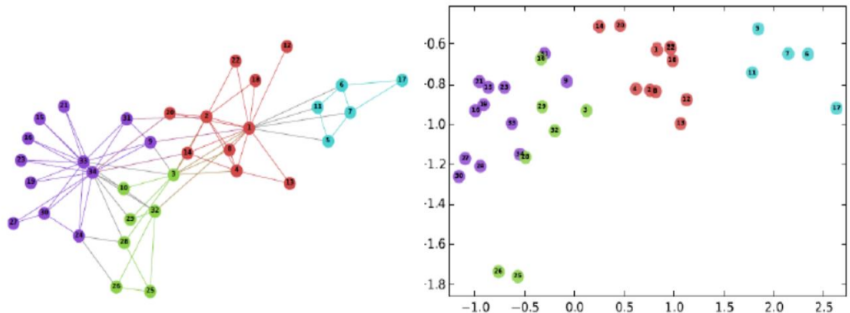
Les méthodes de type DeepWalk factorisent implicitement une matrice dérivée des puissances de transition :

$$M \approx \log\left(\frac{\mathrm{vol}(G)}{T}\sum_{r=1}^{T} P^r D^{-1}\right) - \log K, \qquad P = D^{-1}A.$$

- La fenêtre T agrège plusieurs échelles de diffusion.
- Le logarithme et le décalage dépendent de l'objectif Skip-gram négatif.
- Cette lecture relie marches aléatoires, factorisation et filtrage spectral.

Source : Qiu et al., 2018, Network Embedding as Matrix Factorization.

Graphe de Karaté



Les couleurs indiquent ici des groupes structurels. La visualisation 2D est une projection de l'embedding : elle ne suffit pas à évaluer la qualité dans l'espace de dimension d.

Décodage des embeddings

Classification de nœuds

$$\widehat{y}_v = \mathrm{softmax}(W z_v + b).$$

Évaluer sur des nœuds non utilisés pour ajuster le classifieur.

Prédiction de liens

$$s(u,v) = z_u^\top z_v, \qquad g_{uv} = \mathrm{MLP}([z_u, z_v]),$$

$$s(u,v) = \tfrac{1}{2}(g_{uv} + g_{vu}).$$

Comparer à des paires négatives définies par le protocole.

Pour une relation non orientée, le décodeur doit être symétrique. Pour une arête orientée ou typée, un produit scalaire symétrique est généralement insuffisant.

Embeddings transductifs

DeepWalk et node2vec apprennent un paramètre z_v pour chaque nœud vu à l'entraînement.

- Un nouveau nœud ne possède pas d'embedding sans nouvel ajustement.
- Les attributs de nœuds ne sont pas exploités par la formulation de base.
- Les représentations changent lorsque le graphe évolue fortement.
- La proximité apprise reflète le processus de marche, pas nécessairement la tâche aval.

Les modèles inductifs remplacent la table de vecteurs par une fonction partagée des attributs et du voisinage.

PARTIE V

Message passing et architectures GNN

Principe d'un GNN

À la couche ℓ, chaque nœud porte un état $h_v^{(\ell)}$. Une couche combine

1. les états des voisins ;
2. les attributs d'arêtes éventuels ;
3. l'état courant du nœud.

Après L couches, $h_v^{(L)}$ dépend au plus du voisinage à L sauts.

Induction relationnelle

Les mêmes paramètres sont appliqués à tous les nœuds et à tous les graphes ; la structure détermine où les calculs sont partagés.

Cadre MPNN

Pour chaque couche,

$$m_{uv}^{(\ell)} = M_\ell\Big(h_u^{(\ell)}, h_v^{(\ell)}, e_{uv}\Big),$$

$$m_v^{(\ell)} = \mathrm{AGG}_{u\in\mathcal{N}(v)}\, m_{uv}^{(\ell)}, \qquad h_v^{(\ell+1)} = U_\ell\Big(h_v^{(\ell)}, m_v^{(\ell)}\Big).$$

L'agrégateur doit être invariant à l'ordre des voisins. M_ℓ et U_ℓ peuvent être linéaires, des MLP, des mécanismes d'attention ou des cellules récurrentes.

Source : Gilmer et al., 2017.

Profondeur et champ réceptif

- couche 1 : voisins directs ;
- couche 2 : voisinage à deux sauts ;
- couche L : information jusqu'à L sauts.

Croissance

Dans une approximation arborescente de facteur de branchement b, le voisinage exploré contient de l'ordre de

$$1 + b + \cdots + b^L$$

nœuds avant recouvrements. Le degré moyen seul ne borne pas la taille d'un voisinage particulier.

Une profondeur accrue élargit le contexte mais augmente le coût, la redondance et les phénomènes de lissage ou de compression.

Convolution spectrale localisée

Un filtre spectral s'écrit

$$g_\theta(L)x = U g_\theta(\Lambda) U^\top x.$$

Une approximation polynomiale de degré K,

$$g_\theta(L)x \approx \sum_{k=0}^{K} \theta_k L^k x,$$

est localisée dans un voisinage à K sauts et évite de multiplier explicitement par U.

- Les polynômes de Chebyshev fournissent une approximation stable.
- Le GCN de Kipf–Welling retient une approximation du premier ordre.

Renormalisation dans GCN

Pour un graphe non orienté, on définit

$$\widetilde{A} = A + I, \qquad \widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}, \qquad \widehat{A} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}.$$

Une couche GCN calcule

$$H^{(\ell+1)} = \sigma\Big(\widehat{A}H^{(\ell)}W^{(\ell)}\Big).$$

- $H^{(0)} = X$;
- $\widehat{A}H$ agrège et normalise les voisins ;
- $W^{(\ell)}$ transforme les canaux de caractéristiques.

Source : Kipf et Welling, 2017.

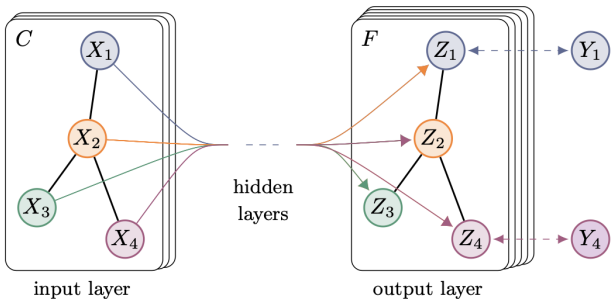
Agrégation dans GCN

Pour un nœud i,

$$h_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{\widetilde{d}_i \widetilde{d}_j}} h_j^{(\ell)} W^{(\ell)} \right).$$

- les boucles propres conservent le signal du nœud ;
- la normalisation limite la domination des nœuds de fort degré ;
- les poids ne dépendent pas du contenu : ils sont imposés par la topologie et les degrés.

Structure de calcul d'un GCN



- entrée : matrice X ;
- propagation : $\widehat{A}H$;
- transformation : multiplication par W ;
- sortie : logits de nœuds ou embeddings intermédiaires.

GraphSAGE

Une formulation moyenne est

$$\bar{h}_{\mathcal{N}(v)}^{(\ell)} = \frac{1}{|S_\ell(v)|} \sum_{u \in S_\ell(v)} h_u^{(\ell)},$$

$$h_v^{(\ell+1)} = \sigma\Big(W^{(\ell)}[h_v^{(\ell)} \| \bar{h}_{\mathcal{N}(v)}^{(\ell)}]\Big),$$

où $S_\ell(v) \subseteq \mathcal{N}(v)$ est un échantillon de voisins.

- les paramètres sont partagés entre nœuds ;
- un nouveau nœud est encodé à partir de ses attributs et voisins ;
- l'échantillonnage borne le coût du voisinage.

Source : Hamilton, Ying et Leskovec, 2017.

Échantillonnage de voisins

Avec des budgets $s_1, \dots, s_L$, le nombre maximal de nœuds chargés par cible est

$$1 + s_L + s_L s_{L-1} + \cdots + \prod_{\ell=1}^{L} s_\ell.$$

Avantages

- mini-batches de nœuds ;
- mémoire contrôlée ;
- entraînement inductif.

Effets statistiques

- variance de l'estimation du voisinage ;
- biais possible selon l'échantillonneur ;
- duplication de nœuds entre arbres de calcul.

Graph Attention Network

Pour une tête d'attention,

$$e_{ij} = \text{LeakyReLU}\left(a^\top [Wh_i \| Wh_j]\right),$$

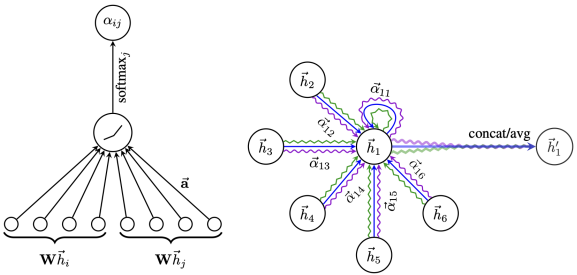
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(e_{ik})},$$

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij} W h_j\right).$$

Les coefficients sont normalisés séparément pour chaque nœud cible i.

Source : Veličković et al., 2018.

Attention locale sur le graphe



- seules les arêtes observées participent au softmax ;
- le score dépend des représentations transformées ;
- les voisins peuvent recevoir des poids distincts ;
- un coefficient élevé n'est pas, à lui seul, une explication causale.

Attention multi-tête

Pour K têtes, les couches cachées concatènent généralement

$$h_i' = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{(k)} W^{(k)} h_j \right).$$

La couche de sortie peut moyenner

$$h_i' = \frac{1}{K} \sum_{k=1}^{K} \sum_{j} \alpha_{ij}^{(k)} W^{(k)} h_j.$$

Chaque tête possède ses propres paramètres ; la concaténation multiplie la dimension de sortie.

Attributs d'arêtes

Un message conditionné par l'arête peut prendre la forme

$$m_{uv}^{(\ell)} = \phi_\ell\Big(h_u^{(\ell)}, h_v^{(\ell)}, e_{uv}\Big),$$

ou, dans une convolution conditionnée,

$$m_{uv}^{(\ell)} = W_\ell(e_{uv}) h_u^{(\ell)}.$$

- Le type, le poids, la distance ou le temps peuvent modifier la transformation.
- Ignorer e_{uv} rend indiscernables des relations pourtant différentes.
- Une base de matrices ou un réseau générateur limite le nombre de paramètres.

Agrégation globale

Après L couches, un readout invariant construit

$$h_G = R\left(\{h_v^{(L)} : v \in V\}\right).$$

Exemples :

$$R_{\text{sum}} = \sum_v h_v, \qquad R_{\text{mean}} = \frac{1}{n}\sum_v h_v, \qquad R_{\text{att}} = \sum_v \beta_v h_v.$$

- la somme conserve l'information de taille ;
- la moyenne normalise la taille ;
- un readout hiérarchique apprend une mise en commun de sous-structures.

Têtes de prédiction

Tâche	**Décodeur**	**Perte typique**
Nœud	$\widehat{y}_v = f(h_v)$	entropie croisée, MSE
Lien	$s_{uv} = g(h_u, h_v, e_{uv})$	logistique avec négatifs, ranking
Graphe	$\widehat{y}_G = f(h_G)$	classification ou régression
Arête	$\widehat{y}_{uv} = f([h_u, h_v, e_{uv}])$	classification multiclasse

La symétrie du décodeur doit suivre celle de la relation : $s(u, v) = s(v, u)$ pour une arête non orientée, mais pas nécessairement pour une arête orientée.

Connexions résiduelles et normalisation

Une couche résiduelle s'écrit

$$H^{(\ell+1)} = \mathrm{Norm}\Big(H^{(\ell)} + F_\ell(H^{(\ell)}, A)\Big).$$

- la branche identité préserve une partie du signal initial ;
- la normalisation contrôle l'échelle des activations ;
- l'ajout de X à plusieurs profondeurs limite la perte d'information locale ;
- la profondeur utile reste déterminée par la portée de la tâche et la structure du graphe.

Complexité d'une couche GNN

Pour $H \in \mathbb{R}^{n \times d_{\text{in}}}$ et $W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, une couche GCN creuse coûte approximativement

$$O(nd_{\text{in}}d_{\text{out}} + md_{\text{out}}).$$

- HW transforme les caractéristiques ;
- $\widehat{A}(HW)$ propage sur les m arêtes ;
- GAT ajoute le calcul des scores d'attention sur chaque arête ;
- le stockage des activations de toutes les couches domine souvent la mémoire d'entraînement.

PARTIE VI

Apprentissage, évaluation et passage à l'échelle

Découpage des données

Tâche	Découpage	Question testée
Nœuds transductifs	labels train/val/test sur un graphe fixe	prédire des labels inconnus du graphe observé
Nœuds inductifs	nœuds ou composantes séparés	généraliser à de nouveaux nœuds
Liens	arêtes positives séparées, négatifs associés	prédire des relations absentes
Graphes	graphes entiers séparés	généraliser à de nouvelles structures
Temporel	entraînement avant une date, test après	prévoir l'évolution future

Fuites de données

Pour la prédiction de liens, les arêtes de validation et de test doivent être retirées du graphe utilisé pour

- calculer les embeddings ;
- construire les voisinages et chemins ;
- calculer les centralités et caractéristiques structurelles ;
- produire les sous-graphes d'entraînement ;
- échantillonner les négatifs : une arête positive réservée au test ne doit pas être traitée comme une non-arête.

Dans un protocole temporel, toute information postérieure à la date de coupure — attribut, arête ou agrégat — appartient au futur.

Échantillonnage négatif pour les liens

Si E^+ contient les liens observés, on construit des paires négatives $E^- \subseteq V \times V \setminus E^+$. Dans un graphe incomplet, « non observé » ne signifie pas nécessairement « négatif » : la distribution choisie définit donc la tâche évaluée.

Négatifs uniformes

- simples à générer ;
- souvent trop faciles dans un graphe creux.

Négatifs difficiles

- mêmes types de nœuds, degrés ou voisinages ;
- plus proches des candidats réellement confondus.

Le ratio $|E^-|/|E^+|$ et la distribution d'échantillonnage doivent être identiques ou explicitement corrigés lors de la comparaison des modèles.

AUC et Average Precision

- **ROC-AUC** : probabilité qu'un positif soit classé devant un négatif tiré selon le protocole.
- **Average precision** : précision moyenne le long de la courbe rappel–précision ; sensible à la prévalence.
- **Hits@k** : présence du lien correct dans les k premiers candidats.
- **MRR** : moyenne de l'inverse du rang du premier candidat correct.

Pour un système de recommandation, l'évaluation par rang face à un ensemble de candidats est généralement plus proche de l'usage qu'une AUC sur des négatifs uniformes.

Régimes de mini-batch

Régime	**Unité du batch**	**Contrainte principale**
Full-batch	graphe complet	mémoire des activations $O(ndL)$
Échantillonnage de voisins	nœuds cibles et arbres locaux	explosion des voisinages
Échantillonnage de sous-graphes	blocs ou partitions	biais de frontière
Batch de graphes	ensemble de petits graphes disjoints	variance des tailles

Les petits graphes sont regroupés par union disjointe : une seule matrice d'adjacence bloc-diagonale suffit au calcul.

Échantillonnage de voisins et de sous-graphes

Voisins par couche

- budget fixe par nœud et par profondeur ;
- adapté à l'inférence centrée sur des nœuds ;
- duplications possibles entre cibles.

Sous-graphe

- nœuds et arêtes mutualisés dans le batch ;
- meilleure localité mémoire ;
- distribution des frontières à contrôler.

Le choix influence l'estimateur de gradient autant que la performance matérielle.

Normalisation des attributs

Pour une caractéristique continue x_j, les statistiques

$$\mu_j, \sigma_j$$

sont estimées sur les nœuds ou graphes d'entraînement, puis appliquées aux partitions suivantes.

- Les catégories inconnues nécessitent une modalité dédiée.
- Les identifiants bruts ne doivent pas être traités comme des nombres ordinaux.
- Les caractéristiques structurelles doivent être recalculées sur le graphe autorisé par le protocole.
- Pour un graphe sans attributs, degré, constantes ou encodages positionnels fournissent un signal initial.

Apprentissage auto-supervisé

Objectif	Signal	Risque
Reconstruction de liens	arêtes observées contre paires négatives	apprentissage du degré ou de la densité
Masquage d'attributs	prédire des caractéristiques cachées	raccourci par corrélation locale
Contrastif	rapprocher deux vues du même objet	augmentations détruisant la sémantique
Contexte / sous-graphe	cohérence locale–globale	négatifs faux ou trop faciles

La fonction de préentraînement doit conserver les invariances utiles à la tâche aval.

Baselines et ablations

Pour une classification de nœuds, comparer au minimum :

- régression logistique ou MLP sur X seul ;
- label propagation si des labels voisins sont disponibles ;
- caractéristiques structurelles suivies d'un modèle tabulaire ;
- embedding de marches aléatoires ;
- GCN, GraphSAGE ou GAT avec budget comparable.

Les ablations X seul, A seul et $X + A$ identifient la contribution respective des attributs et de la topologie.

Hyperparamètres graphiques

Architecture

- profondeur et dimension cachée ;
- agrégateur et normalisation ;
- connexions résiduelles ;
- nombre de têtes.

Échantillonnage

- nombre de voisins par couche ;
- taille des sous-graphes ;
- négatifs par positif ;
- distribution des négatifs.

La sélection s'effectue sur validation avec une métrique alignée sur le test et l'usage cible.

Variabilité expérimentale

Rapporter :

- la définition exacte des partitions et des négatifs ;
- plusieurs graines pour initialisation, batchs et échantillonnage ;
- moyenne, dispersion et nombre de répétitions ;
- nombre de paramètres, mémoire, temps d'entraînement et d'inférence ;
- prétraitements du graphe, suppression d'arêtes et normalisations ;
- meilleur epoch choisi sur validation, sans sélection sur le test.

Sur les petits benchmarks, la variance entre splits peut dépasser le gain annoncé entre architectures.

PARTIE VII

Expressivité, limites et extensions

Homophilie et hétérophilie

Une mesure simple pour des labels discrets est

$$h_{\text{edge}} = \frac{1}{m} \sum_{(u,v)\in E} \mathbf{1}[y_u = y_v].$$

Homophilie

Les voisins ont souvent des labels ou attributs similaires : moyenne et diffusion sont adaptées.

Hétérophilie

Les arêtes relient des rôles différents : agréger sans distinguer les types peut effacer le signal.

Le degré d'homophilie doit être mesuré par rapport à la cible et comparé à un modèle nul tenant compte des fréquences de classes.

Sur-lissage

Dans un GCN linéarisé à matrices de poids compatibles,

$$H^{(L)} = \widehat{A}^L X \left(\prod_{\ell=0}^{L-1} W^{(\ell)} \right).$$

Lorsque L augmente, les composantes associées aux valeurs propres de module inférieur à 1 s'atténuent ; les représentations convergent vers un sous-espace de faible dimension.

- les nœuds deviennent difficilement séparables ;
- le phénomène dépend du spectre et de la normalisation ;
- résidus, injection de X, régularisation et profondeurs adaptées retardent cette convergence.

Sur-écrasement

Le nombre de nœuds à distance L peut croître exponentiellement, alors que $h_v^{(L)} \in \mathbb{R}^d$ conserve une dimension fixe.

- les dépendances lointaines doivent traverser des coupes étroites ;
- de nombreux messages sont agrégés dans le même canal ;
- le gradient entre nœuds éloignés peut devenir très faible.

Augmenter la largeur, modifier le graphe, introduire des raccourcis ou des mécanismes globaux sont des réponses distinctes à ce goulot d'étranglement.

Source : Alon et Yahav, 2021 ; Topping et al., 2022.

Expressivité et test 1-WL

Le raffinement de Weisfeiler–Leman met à jour une couleur par

$$c_v^{(\ell+1)} = \mathrm{HASH}\left(c_v^{(\ell)}, \left\{\!\left\{ c_u^{(\ell)} : u \in \mathcal{N}(v) \right\}\!\right\}\right).$$

Un MPNN fondé sur une agrégation multiensemble ne distingue pas deux graphes que 1-WL ne distingue pas.

- certains graphes réguliers non isomorphes restent indiscernables ;
- l'expressivité dépend de l'injectivité de l'agrégateur et de la mise à jour ;
- davantage de profondeur ne dépasse pas cette borne structurelle.

Source : Xu et al., 2019 ; Morris et al., 2019.

Graph Isomorphism Network

La couche Graph Isomorphism Network est

$$h_v^{(\ell+1)} = \mathrm{MLP}_\ell \left((1+\epsilon_\ell) h_v^{(\ell)} + \sum_{u \in \mathcal{N}(v)} h_u^{(\ell)} \right).$$

- la somme peut distinguer des multisets que la moyenne confond ;
- sous les hypothèses du résultat théorique (multisets bornés sur un domaine dénombrable), la somme suivie d'un MLP peut représenter une mise à jour injective ;
- le readout concaténant plusieurs profondeurs conserve plusieurs échelles.

Atteindre la borne 1-WL ne signifie pas distinguer tous les graphes non isomorphes.

Source : Xu et al., 2019.

Encodages positionnels

Pour un graphe connexe, les k premiers vecteurs propres non triviaux du Laplacien fournissent

$$P = [u_2, \ldots, u_{k+1}] \in \mathbb{R}^{n \times k}.$$

Ils peuvent être concaténés aux attributs :

$$H^{(0)} = [X \| P].$$

- ils encodent des coordonnées globales liées à la diffusion ;
- le signe de chaque vecteur propre est indéterminé ;
- les sous-espaces associés à des valeurs propres multiples peuvent tourner ;
- les distances de plus court chemin et temps de marche constituent d'autres encodages structurels.

Graphes hétérogènes

Soient des types de nœuds $\tau(v)$ et de relations $r(u, v)$. Une couche relationnelle peut utiliser

$$h'_v = \sigma\left(W_0 h_v + \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{N}_r(v)} \frac{1}{c_{v,r}} W_r h_u\right).$$

- W_r distingue les sémantiques des relations ;
- une décomposition en bases réduit le coût quand $|\mathcal{R}|$ est grand ;
- les types de nœuds peuvent avoir des espaces d'attributs différents.

Graphes temporels

Une interaction est un triplet (u, v, t, e_{uvt}). Un modèle temporel combine

$$h_v(t^-), \quad h_u(t^-), \quad e_{uvt}, \quad \Delta t$$

pour mettre à jour l'état après l'événement.

- la causalité interdit l'utilisation d'arêtes postérieures à t ;
- la mémoire du nœud résume son historique ;
- l'échantillonnage de voisins doit respecter le temps ;
- l'évaluation suit une coupure chronologique et des candidats disponibles à la date considérée.

Transformers de graphes

Un Transformer appliqué à tous les nœuds calcule

$$\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}} + B_G\right)V,$$

où B_G encode la structure : distances, arêtes, degrés ou positions spectrales.

- l'attention globale réduit la contrainte de portée locale ;
- son coût dense est $O(n^2 d)$;
- sans encodage B_G, la structure du graphe n'est pas explicitement distinguée ;
- les variantes creuses limitent l'attention à des motifs ou voisinages sélectionnés.

Choix du modèle

Régime	Point de départ	Extension pertinente
Topologie fixe, peu de labels	propagation, méthodes spectrales	GCN transductif
Nouveaux nœuds fréquents	GraphSAGE	échantillonnage de sous-graphes
Relations pondérées ou typées	MPNN avec attributs d'arêtes	R-GCN, convolution conditionnée
Interactions lointaines	encodages positionnels	rewiring ou attention globale
Hétérophilie forte	modèle séparant les types de voisins	filtres signés ou multi-canaux
Graphes dynamiques	découpage temporel	mémoire et messages horodatés

PARTIE VIII

Références essentielles

Références : graphes, diffusion et méthodes spectrales

- U. von Luxburg, *A Tutorial on Spectral Clustering*, Statistics and Computing, 2007.
- S. Brin et L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Computer Networks and ISDN Systems, 1998.
- D. Zhou et al., *Learning with Local and Global Consistency*, NeurIPS, 2003.
- X. Zhu, Z. Ghahramani et J. Lafferty, *Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions*, ICML, 2003.
- M. E. J. Newman, *Networks: An Introduction*, Oxford University Press, 2010.
- V. D. Blondel et al., *Fast Unfolding of Communities in Large Networks*, Journal of Statistical Mechanics, 2008.
- U. Brandes, *A Faster Algorithm for Betweenness Centrality*, Journal of Mathematical Sociology, 2001.

Références : embeddings et représentation

- B. Perozzi, R. Al-Rfou et S. Skiena, *DeepWalk: Online Learning of Social Representations*, KDD, 2014. arXiv:1403.6652
- A. Grover et J. Leskovec, *node2vec: Scalable Feature Learning for Networks*, KDD, 2016. arXiv:1607.00653
- J. Qiu et al., *Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec*, WSDM, 2018. arXiv:1710.02971
- A. Ahmed et al., *Distributed Large-scale Natural Graph Factorization*, WWW, 2013.
- W. L. Hamilton, R. Ying et J. Leskovec, *Representation Learning on Graphs: Methods and Applications*, 2017. arXiv:1709.05584

Références : GNN, expressivité et limites

- T. N. Kipf et M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, ICLR, 2017. arXiv:1609.02907
- W. L. Hamilton, R. Ying et J. Leskovec, *Inductive Representation Learning on Large Graphs*, NeurIPS, 2017. arXiv:1706.02216
- J. Gilmer et al., *Neural Message Passing for Quantum Chemistry*, ICML, 2017. arXiv:1704.01212
- P. Veličković et al., *Graph Attention Networks*, ICLR, 2018. arXiv:1710.10903
- K. Xu et al., *How Powerful are Graph Neural Networks?*, ICLR, 2019. arXiv:1810.00826
- U. Alon et E. Yahav, *On the Bottleneck of Graph Neural Networks*, ICLR, 2021. arXiv:2006.05205
- J. Topping et al., *Understanding Over-squashing and Bottlenecks on Graphs via Curvature*, ICLR, 2022. arXiv:2111.14522