

Analyse et apprentissage de graphes

un graphe est une collection de nœuds, avec des connexions

exemple :

- egonetwork (dimRedIn)
- réseaux électriques,
- réseaux de protéines,
- réseaux internet.

Plusieurs objectifs :

- Analyse de graphe : mesurer et quantifier les réseaux, détecter des communautés, identification de nœuds importants.
- Apprentissage de graphe : prédiction de liens, classification de nœuds, embedding de graphes.

* Notions de bases sur les graphes

= ensemble de nœuds et d'arêtes.

$$G = (V, E)$$

vertices
nœuds
 $\{1, \dots, n\}$

$E \subseteq V \times V$

Une arête $(i, j) \in E$ = lien entre i et j \Rightarrow les nœuds sont adjacents ou voisins.

Degré du nœud = nombre de voisins du nœud.

Un graphe complet : il y a une arête entre toutes les paires de nœuds (\Rightarrow degré $n-1$).

Un chemin = séquence des arêtes entre i et j .

Un cycle = chemin qui commence et finit au même nœud.

Longueur du chemin = nombre d'arêtes sur le chemin.

Chemin géodésique = le plus court chemin entre i et j .

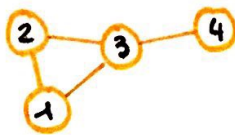
Diamètre d'un graphe = longueur du plus long des plus courts chemins entre deux nœuds.

Graphe connecté = s'il a une seule composante connexe
il existe un chemin entre tous les nœuds.

Graphe non dirigé = pas de sens dans les arêtes.

Représentation en mémoire

liste d'arêtes

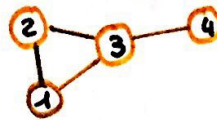


1 2
1 3
2 3
3 4

ou liste d'adjacence

1: 2 → 3
2: 1 → 3
3: 1 → 2 → 4
4: 3

Matrice d'adjacence



$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

symétrique

⇒ la meilleure représentation dépend de la mémoire dispo et de ce que l'on veut stocker.

* Analyse de graphes.

1- Mesure globale de réseaux

⇒ il existe 2 mesures descriptives que l'on peut illustrer avec deux modèles de graphes aléatoires.

Erdős - Rényi

Deux paramètres:

- le nb de nœuds n
- probabilité p .

⇒ ajout d'arêtes avec probab de façon indépendante, aléatoire uniforme.

Berabasi - Albert

Deux paramètres:

- Un graphe initial avec n nœuds
- probabilité p .

⇒ ajout de nœuds de manière séquentielle, connexion d'un nœud à plusieurs nœuds avec probab p .

(Attachement préférentiel)

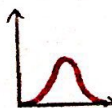
un nœud qui a beaucoup de connexions va attirer plus de connexions.

⇒ mesure qui permettent de discerner ces graphes?

Degré de distribution

P_R = proba qu'un nœud aléatoire ait un degré k

bimomiale


$$P_R = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

- concentrée autour de la moyenne
- décroît exponentiellement vite

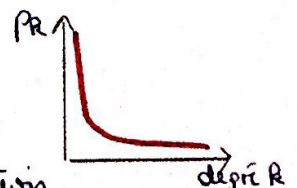
degré moyen = np max = moyenne

loi de puissance

$$P_R \propto R^{-\alpha}$$

- Heavy tailed distribution
- Scale free: degré moyen non informatif

degré moyen = une constante si $\alpha \geq 2$
maximum = $O(n^{\frac{1}{\alpha-1}})$



Coefficient de clustering

=> mesure qui dit à quel point les nœuds tendent à former un cluster ensemble.

Coefficient local

$$C_i = \frac{\text{triangles centrés au nœud } i}{\text{triplet centré au nœud } i}$$

A quel point un nœud i et ses voisins sont proches de former un graphe complet.

Coefficient global

$$CC = \frac{1}{n} \sum_{i=1}^n C_i$$
 densité des triangles dans le graphe.

Erdős-Rényi

$$E(CC) = E(C_i) = p$$

probabilité de 2 voisins d'un nœud d'être aussi voisins indépendamment de la struct local

Barabási-Albert

CC suit approximativement une loi de puissance

$$C(k) = k^{-1} \Rightarrow \text{indique une structure hiérarchique.}$$

Coefficient de clustering moyen.

2- Détection de communauté

=> partitionner le graphe en un groupe de cluster selon un certain critère de qualité

Bonne communauté = un ens de nœuds où les nœuds du m cluster sont très connectés entre eux.
et/ou peu connectés aux nœuds extérieurs.

1) Définir un critère de qualité ...

... basé sur des connexions internes ...

- Densité interne des arêtes $\frac{m_s}{n_s(n_s-1)/2}$
- Degré moyen interne $\frac{2m_s}{n_s}$

ou externes.

- Expansion $\frac{o_s}{n_s}$ - nombre de nœuds dans S
- Ratio cut $\frac{o_s}{n_s(n-n_s)}$ - nombre de nœuds dans le graphe

... basé sur les deux

- Conductance
- Normalized cut
- Modularité

$$\frac{o_s}{2m_s + o_s} + \frac{o_s}{2(m-m_s) + o_s}$$

nombre d'arêtes entre S et V/S

$$\frac{1}{4} (m_s - E(m_s))$$

nombre d'arêtes dans S

nombre d'arêtes dans le graphe

Modularité

=> l'espérance $E(m_s)$ est calculée par rapport à un processus aléatoire qui préserve le degré de chaque nœud.

- chaque arête est divisée en 2 parties.
- chaque partie est combinée à une autre aléatoirement.

$$\frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \mathbb{I}_{\{c_i = c_j\}}$$

=> permet de comparer le nb d'arête que je vois avec le nb d'arête que je m'attendrais à voir si la communauté avait été choisie arbitrairement.

On va chercher à maximiser la modularité (requiert de considérer un nb de groupes exponentiel. (très coûteux).

② Choisir un algorithme pour trouver la communauté qui optimise un critère.

Méthode de Louvain

Init = au début chaque nœud a une communauté

Ensuite, ... l'algo alterne entre deux phases jusqu'à convergence

• optimiser la modularité locale

on regarde les voisins du nœud et si on améliore la modularité quand on merge le nœud avec les voisins, alors on garde ce merge.

• créer un nouveau graphe pondéré

on pondère ensuite chaque communauté par le nombre de liens dans la communauté

et chaque lien entre les paires de communautés par le nombre de liens entre ces communautés.

=> algorithme glouton car décision locale, on fait le travail pour chaque nœud.

=> une fois qu'on a merge, on ne met pas en cause les choix donc algo efficace et optimale.

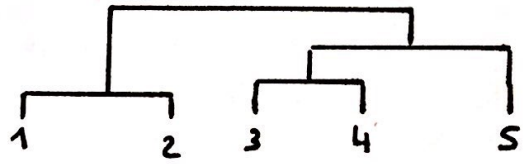
=> obtention d'une solution approchée (on a pas de garanties).

* Autre approche: Clustering hiérarchique.

- => analyse de la structure de communauté à + échelles.
- => construire une hiérarchie de clusters.

Approche bottom-up

= on commence avec un cluster à chaque nœud (comme leurrain)



À chaque itération, on merge les deux clusters les plus proches
=> algo glouton.

=> Il faut définir une notion de dissimilarité entre nœuds et entre ensemble de nœuds.

Distances

Une distance naturelle est celle du plus court chemin entre deux nœuds i et j (on mettra les distances dans une matrice de dissimilarité).

exemples:

- Minimum linkage $D(C_1, C_2) = \min_{i \in C_1, j \in C_2} d(i, j)$
- Average linkage $D(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d(i, j)$

3-Identification de nœuds importants

=> ordonner les nœuds selon une mesure de centralité (= importance)

Notion de marche dans un graphe = chemin qui peut passer plusieurs fois par le même nœud.

Mesures de centralité = tourne autour de l'idée de marche et varie selon le type de marche considérée et la manière de les compter.

Centralité de degré

$$C(x_i) = d_i$$

Nombre de marche de longueur 1 finissant au nœud i

Centralité de vecteurs propres

$$C(x_i) = \frac{1}{\lambda} \sum_{j=1}^n A_{ij} C(x_j)$$

λ est val propre de A
Nombre de marches de longueur ∞ finissant au nœud i .
Important nœuds liés à une page importante).

Closeness centrality

$$C(x_i) = \frac{1}{\sum_{j \neq i} d(i, j)}$$

Inversement proportion à la somme des longueurs des plus courts chemins aux autres nœuds.

Betweenness centrality

$$C(x_i) = \sum_{i \neq j \neq k} \frac{\sigma_{j,k}(i)}{\sigma_{j,k}}$$

Nombre de fois que le nœud agit comme un pont entre deux nœuds.

* Apprentissage de graphes

→ on veut faire des prédictions à partir des graphes.

Prédiction de liens

⇒ prédire de nouvelles arêtes qui peuvent être :

- des interactions futures
- des arêtes manquantes

Approche standard : on utilise la structure en réseau

⇒ utilisation d'une mesure de similarité entre les paires de nœuds pour ranker les arêtes.

(les top rank sont les plus probablement correctes).

exemple de mesures :

Coefficient de Jaccard.

$$S(i, j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$$

⇒ prédiction des top-k arêtes ou utilisation d'un seul

⇒ enrichissement possible par des info de communauté

Evaluation :

- cacher un ens de nœuds et prédire le reste du graphe
- graphe dense : proportion de prédictions correctes.
- graphe sparse : AUC

$$AUC(s) = \frac{1}{|E^+||E^-|} \sum_{e^+ \in E^+} \sum_{e^- \in E^-} \mathbb{I}\{s(e^+) > s(e^-)\}$$

Node Labelling.

Approche semi supervisé

⇒ prédire des labels de nœuds manquants.

Hypothèse principale : smoothness.
2 voisins liés dans le graphe vont avoir tendance à avoir le même label.

la propriété "smoothness" de la fonction graphe est donnée par la forme quadratique de la laplacienne.

$$S_G(f) = f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

(matrice laplacienne est PSD et est égale à $D - W$ ⇒ nous donne des degrés poids

info sur les composantes connexes).

Si $S_G(f)$ est petit, f ne varie pas trop dans les régions denses du graphes.

Manifold : espace défini comme des variétés (SEV d'un euclidien)

⇒ homeomorphe.

le graphe construit va nous permettre de retrouver une structure globale à partir de voisinages.

Deux points sont reliés si ils sont assez similaires (localité).

⇒ approche la structure Manifold

ex algo : k plus proches voisins, ϵ neighborhood...

on renforce cet algo par une regularisation Manifold qui est une approximation de

$$\|f\|_F^2 : \|f\|_F^2 \approx \frac{1}{n^2} f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

ex: laplacian svn.